The MagPi

Issue 90 | February 2020 | magpi.cc

The official Raspberry Pi magazine

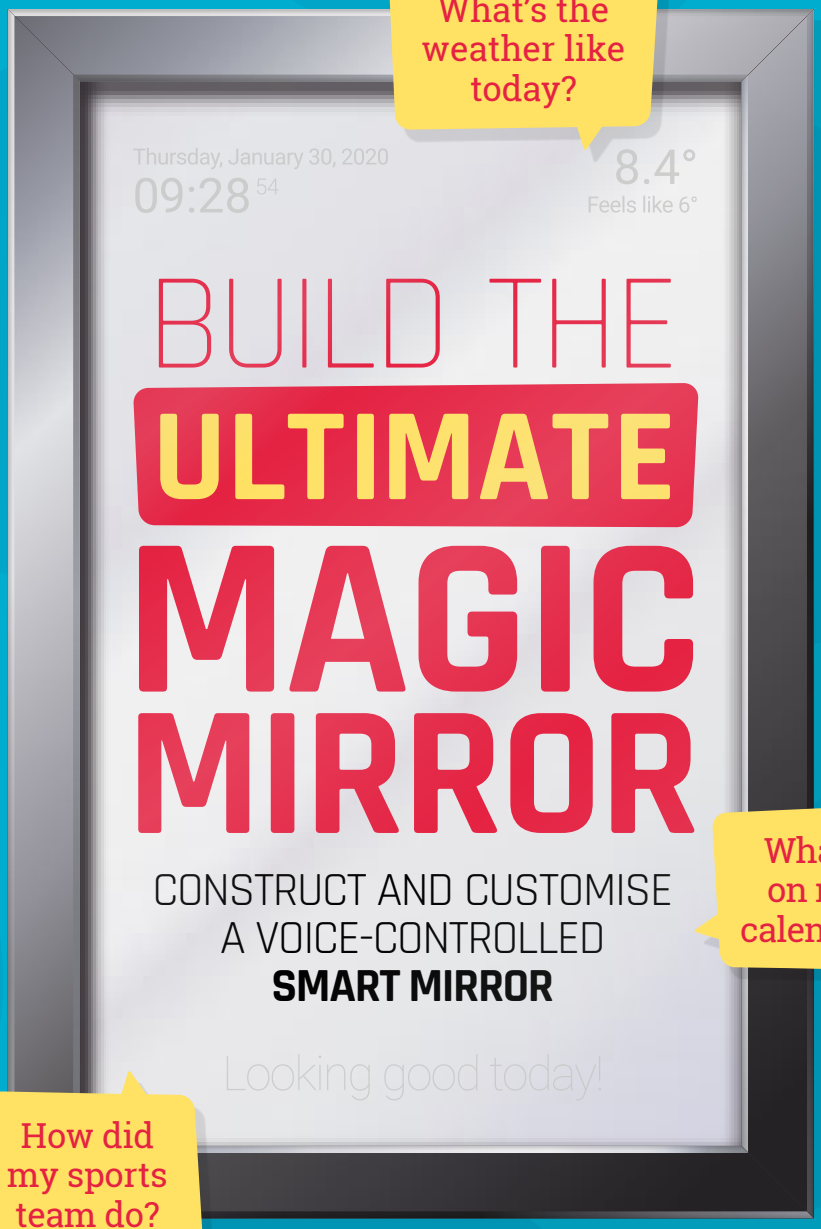
FREE GIFT!

COOLING STAND

KEEP YOUR RASPBERRY PI 4 COOL WITH THIS VERTICAL STAND!

STAND NOT INCLUDED WITH DIGITAL EDITION

What's the weather like today?



BUILD THE ULTIMATE MAGIC MIRROR

CONSTRUCT AND CUSTOMISE A VOICE-CONTROLLED **SMART MIRROR**

What's on my calendar?

How did my sports team do?

OVERCLOCK RASPBERRY PI 4

Run at 2.1GHz with a 750MHz GPU

COOLING CASES TESTED!

Discover the best cooling solution

NEW YEAR, NEW PROJECTS!

Build something amazing in 2020

+

LEARN PYTHON

3D PRINT A KEYRING

SET UP AN ALARM BOX

BUILD A BEAT-BOX

£5.99

40 PAGES OF PROJECTS & GUIDES

WELCOME
to *The MagPi* 90

EDITOR
Lucy Hattersley

Lucy is editor of *The MagPi*. This month she did Veganuary and dry January. She'll see you at the bar in February.

magpi.cc

By now you will have noticed the free gift on the print edition of *The MagPi*. You may be wondering how useful a single strip of plastic can be? Very, very useful as it happens.

You see, back in issue 88, Gareth Halfacree tested a Raspberry Pi 4 with all the different firmware updates. This gave us an insight into just how important firmware was to the Raspberry Pi. But another thing came to light: Raspberry Pi 4 runs cooler when stood vertically.

Alex Bate, Raspberry Pi's digital content manager, got to work on a design that's now attached to the front cover. All is explained on page 6.

Do yourself a favour and unstick the Raspberry Pi 4 stand, then pop your Raspberry Pi 4 on its side and keep it vertical.

It turns out that keeping Raspberry Pi 4 cool enables it to run at full speed for longer and, crucially, this opens up a world of overclocking, where you try to squeeze ever more performance out of a Raspberry Pi. Our overclocking tutorial (page 34) explains all.

Raspberry Pi 4 has always been cool to us. But cool is fast in the world of computing, and this month we're all about running Raspberry Pi as fast as possible.

Lucy Hattersley Editor

GET A RASPBERRY ZERO W KIT WITH A SUBSCRIPTION! PAGE 22

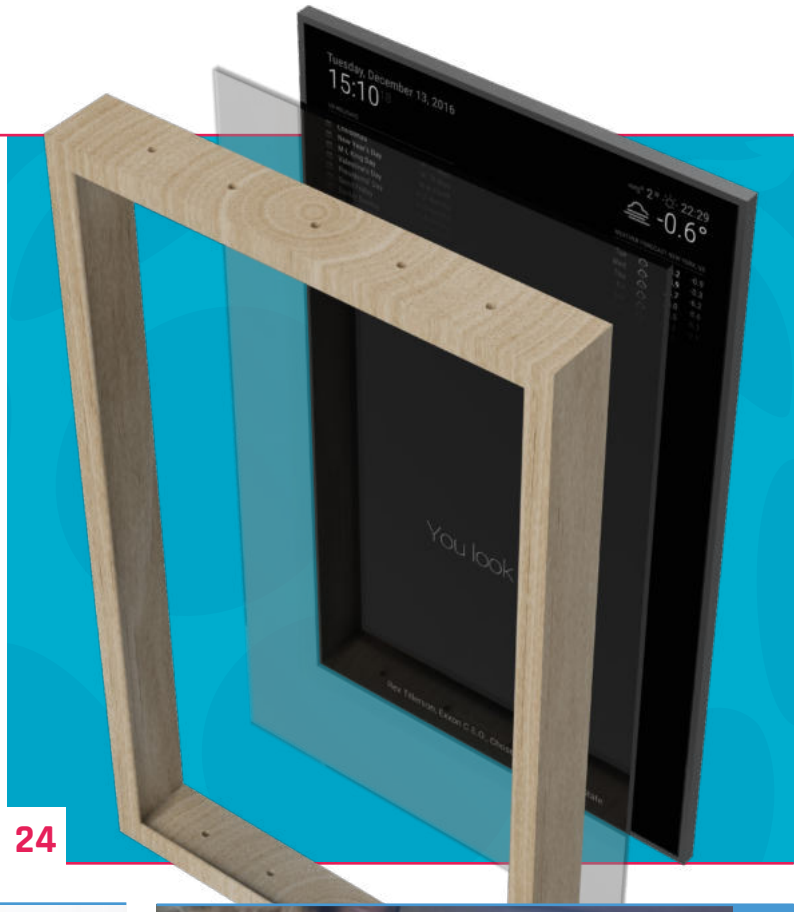
Contents

Issue 90 • February 2020

Cover Feature

Regulars

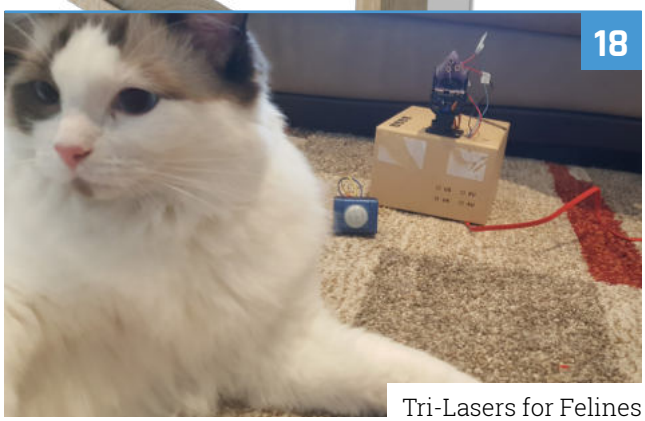
Project Showcases



24

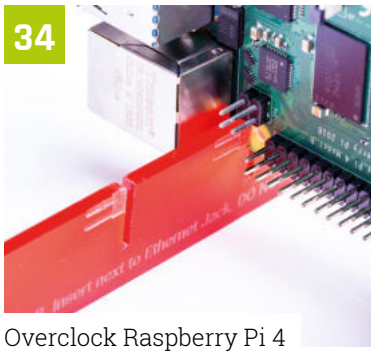
20

Raspberry Pi Recovery Kit

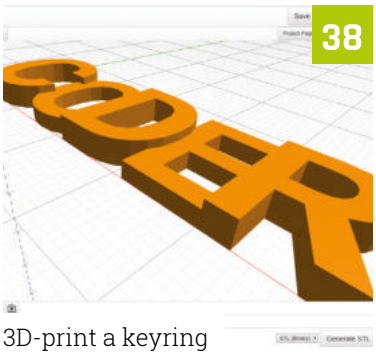


18

Tri-Lasers for Felines



Overclock Raspberry Pi 4



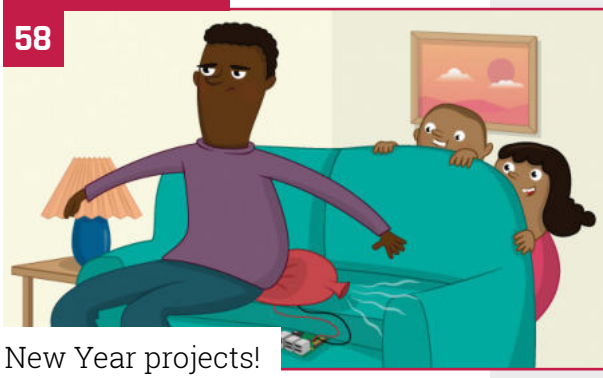
3D-print a keyring

Thermal cases group test

Pranjali Pathak interview

Tutorials

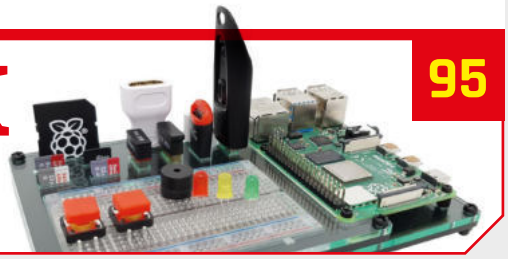
The Big Feature



Reviews

Community

WIN ONE OF FIVE **PROTODOCK** ADD-ONS!



Raspberry Pi 4 Stand

Heat is the enemy of computer efficiency, which is why a stand to keep Raspberry Pi 4 cool is a big deal. **Rosie Hattersley** meets its maker

There's a curious piece of plastic proudly affixed to the front cover of this month's print edition of *The MagPi* magazine. The Raspberry Pi 4 Stand is an exclusively designed holder that does wonders for your computer's efficiency. The stand holds up to three Raspberry Pi 4 computers at once, and has been met with open arms by overclocking maestros.

Issue 88 of *The MagPi* magazine included a thermal testing feature conclusively showing that running a Raspberry Pi 4 on its side enables it to run cooler, and therefore at full speed, for longer.

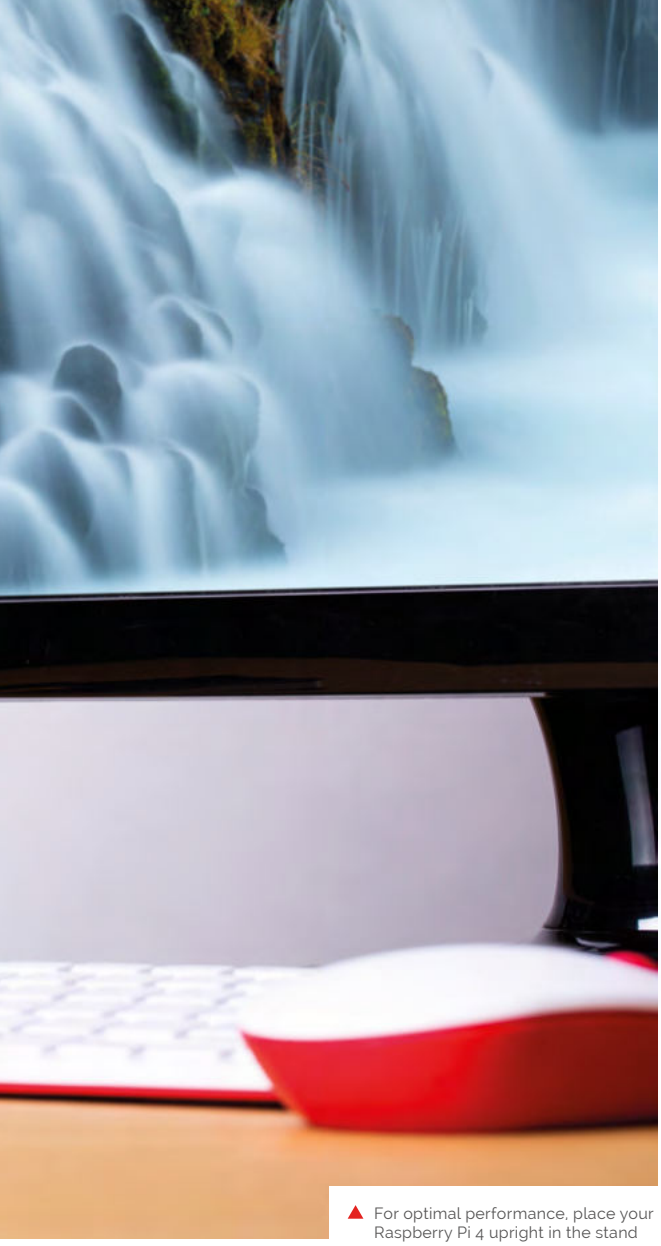
To try out this principle for yourself, turn to this issue's overclocking tutorial on page 34.

If you're reading the digital version of this article, you'll be able to get your hands on your own Raspberry Pi 4 Stand. We're making the files available for you to 3D-print or laser-cut your stand: find them on our GitHub page at **magpi.cc/github**.

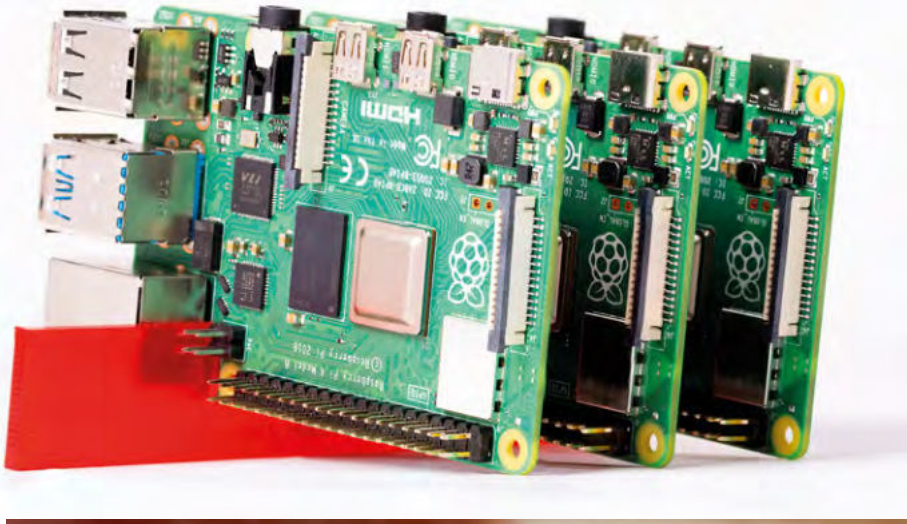
Making a stand

Alex Bate is Raspberry Pi's digital content manager and the stand's inventor. "We were keen to design a system for securing Raspberry Pi on its side, allowing for greater airflow around the device," she tells us. This increased airflow leads to improved cooling efficiency, which enables Raspberry Pi 4 to run optimally and makes it more receptive to overclocking.

"The original idea was the design itself," says Alex. "It's so simple that you could cut slots in a piece of cardboard and recreate the stand at home. We wanted to make the system as accessible as possible, [so offering it as a gift



▲ For optimal performance, place your Raspberry Pi 4 upright in the stand



▲ Instructions printed on the stand itself indicate which way Raspberry Pi slots in

> You'll be surprised how cool your Raspberry Pi 4 runs

with] the magazine was the obvious choice. The idea of a gift, and the idea of having something this simple to recreate came hand in hand."

This, in turn, led to the idea of making it a one-piece stand that was easy to cover-mount on the magazine, and easy for readers to reproduce with a 3D printer.

Made to measure

Alex began playing around with lengths and taking a lot of measurements of numerous Raspberry Pi 4 boards to determine the correct width for the slots (silicon can vary by a fraction of a millimetre). Eventually, the final design came to life.

The stand has small indentations on one side, which guide around the four Power over Ethernet (PoE) pins on one side of Raspberry Pi 4. "I was keen on making it as fool-proof as possible," explains Alex, "and adding the PoE silhouette, I feel, makes it pretty obvious how you should use it."

Alex says if you place a single Raspberry Pi in the middle slot, you'll be surprised how cool your Raspberry Pi 4 runs. You can also build a small cluster with three perfectly cool Raspberry Pi boards.

WeatherMan

MAKER

Martin Mander

Martin works in the Intelligence & Analytics Department at Norfolk County Council. He's passionate about giving old tech a new lease of life, especially projects involving Raspberry Pi.

magpi.cc/oldtechprojects

A shiny music box with a clear door for a screen proves an ideal upcycling candidate for an eighties tech fan keen to keep tabs on the weather. **Rosie Hattersley** hears more

Martin Mander is no stranger to repurposing outdated tech. "I started converting and upcycling vintage technology about ten years ago, after a brief stint blogging for Wired's GeekDad column," he tells us. Discovering he enjoyed documenting projects as much as building them, he began specialising in giving new purpose to broken old tech – "especially items I remember being 'the latest thing' during the 1970s and 1980s when I was a child."

Martin became enamoured of all things Raspberry Pi-flavoured in 2014. Looking for a one-box alternative to a PC for converted gadgets, he was tipped off by a reader that Raspberry Pi might work. He duly made a Raspberry Pi-powered VCR. Numerous 'fruitful' projects followed. There's now a sizeable archive at **magpi.cc/oldtechprojects**.

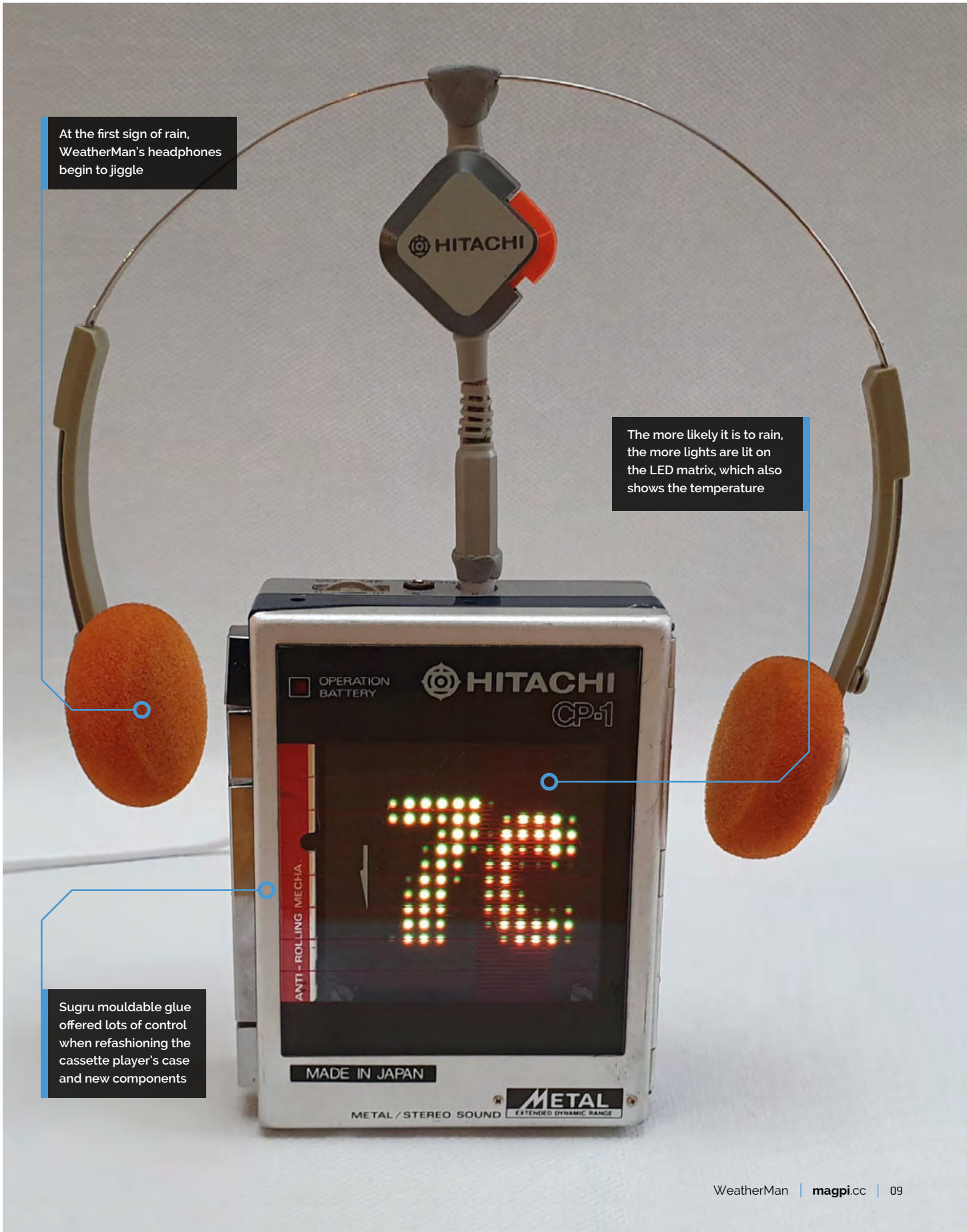
The Walkman, of course, fits that description to a tee. Sony alone sold 200 million, while other brands of cassette player were also legion.

> "I immediately fell in love with it... It had a great retro look"

Press play

Martin's chance to 'rescue' one came when his mother-in-law sent him a broken old Hitachi player that she found during a clear-out.

"I immediately fell in love with it," Martin recalls. "It had a great retro look that drew me in". On closer examination he realised the window in


At the first sign of rain, WeatherMan's headphones begin to jiggle
The more likely it is to rain, the more lights are lit on the LED matrix, which also shows the temperature
Sugru mouldable glue offered lots of control when refashioning the cassette player's case and new components
HITACHI
OPERATION
BATTERY
HITACHI
CP-1
ANTI - ROLLING MECHA.
MADE IN JAPAN
METAL / STEREO SOUND
METAL
EXTENDED DYNAMIC RANGE

Quick FACTS

- Martin's first tech encounter was with a BBC Micro at school
- He half expected singers to appear if music cassettes were used with it
- Martin's home has 25 Raspberry Pi devices for both security and entertainment duties
- He has a 'lovely but trashed' 1960s Dansette radio ripe for conversion…
- Key attraction: it perfectly matches the colour scheme in his bathroom

▲ The early 1980s-style ambient IoT weather display has a current conditions animation

> "I'd been looking for a small case to hold a weather display for my desk for a while"

the cassette player's door was – to the millimetre – the same size as a Raspberry Pi HAT. "At that point, all my other projects hit the back burner," he says.

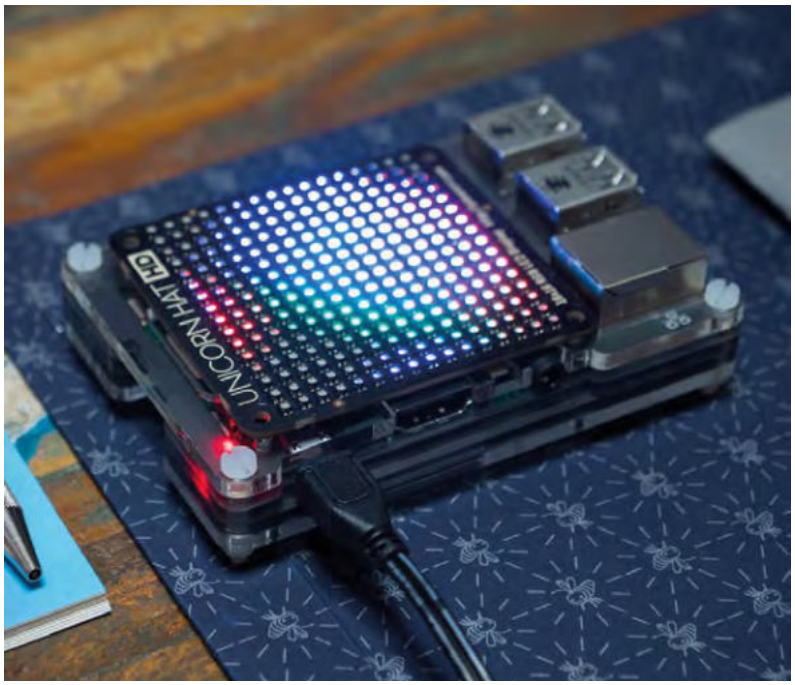
"I'd been looking for a small case to hold a weather display for my desk for a while, and this was the perfect thing. I also had an unused Unicorn HAT HD lying around and this seemed like the ideal project for it." To this he added servos and an awful lot of Sugru – a sort of malleable glue.

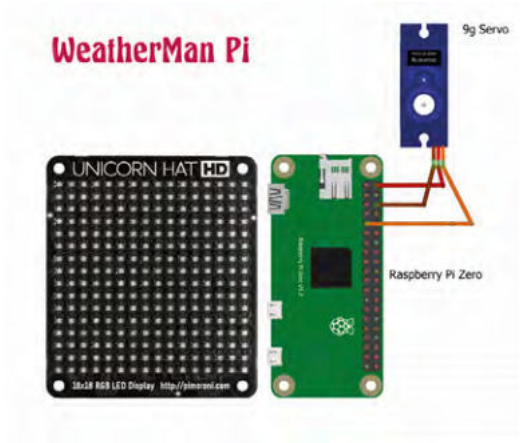
"In the past I've used Lego and Meccano to put things together, but with space being so tight in the case, I used Sugru this time around. It's very workable but sets hard like plastic, perfect for fixing components in the right place."

The WeatherMan currently relies on API data from the Dark Sky weather service, but Martin eventually hopes to link it up to a Raspberry Pi weather station. "With a Raspberry Pi at each end, it should hopefully be straightforward!" he says optimistically.

Getting the details right

The main aim for what became the WeatherMan project was to keep the exterior as true as possible to the original. He wanted it to look like an





▲ Helpfully, a Raspberry Pi Unicorn HAT is precisely the same size as the cassette player's display window

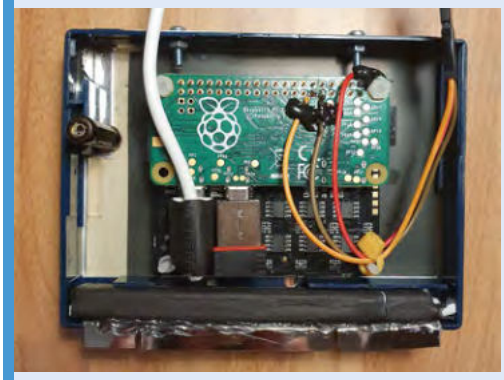
◀ Diagram showing the relatively simple setup involved in creating the WeatherMan project

ornamental piece sitting on his desk speaker, hiding a useful IoT (Internet of Things) device until it bursts into life.

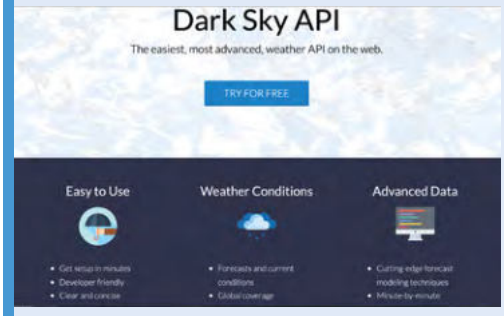
Martin thought the tape player seemed a bit naked without the iconic eighties headphones, so he looked for ways to incorporate them into the build. He drilled out the original jack plug and fittings and joined them together using a 2 mm threaded rod. With a small nut on each end of the rod and a servo connector at the bottom, the headphones now respond whenever the weather is about to change. Their servo-controlled jiggling always makes Martin smile, even if it's to alert him rain is on the way.

Make your own WeatherMan

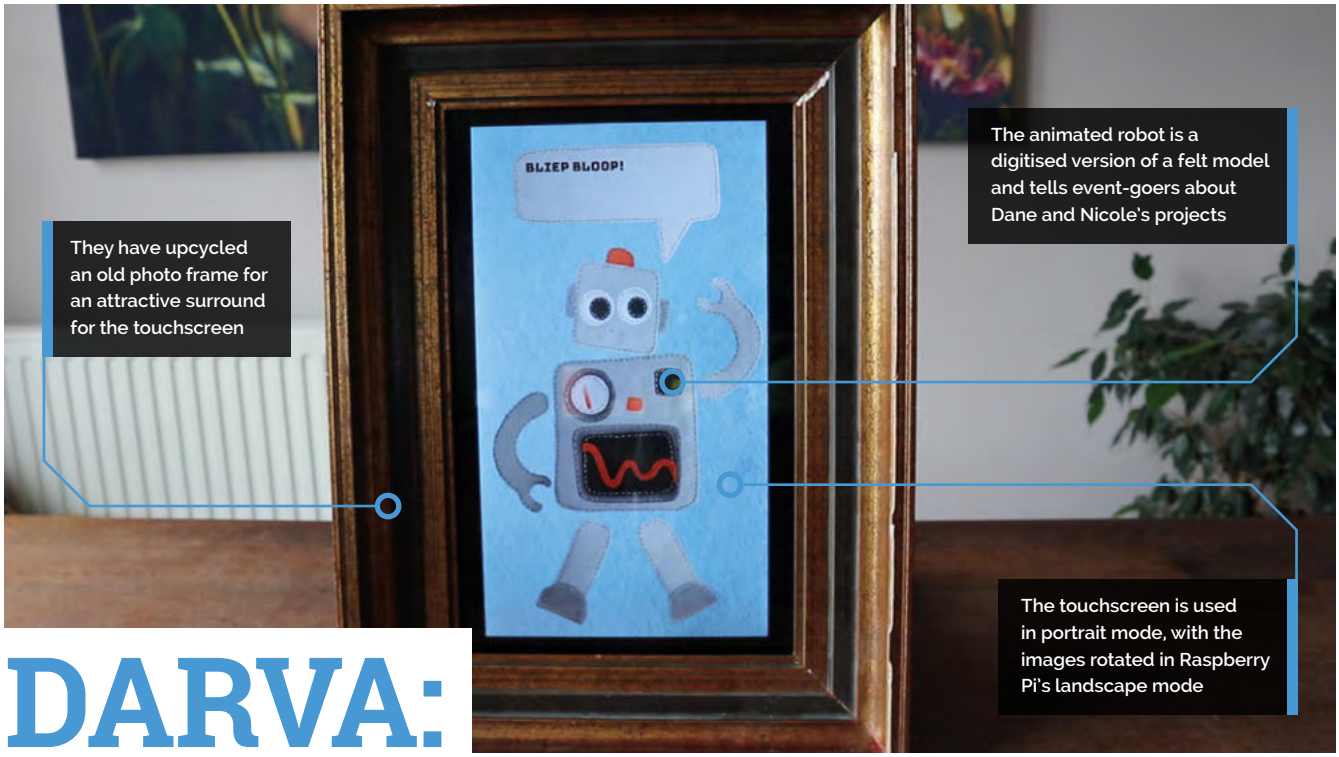
01 Dismantle the cassette player to create as much space as you can inside. Retain any parts you want to reuse. You may need to trim down the case inside to make the components fit nicely.



02 You may need new bolts or a tiny magnet to make the cassette tape door close properly once you've tinkered inside. When you're satisfied that it works well, hot-glue the cassette player case and door.



03 Use Raspbian Buster and adapted Python scripts to retrieve weather data from Dark Sky, display info on the Unicorn HAT, and (optionally) jiggle the headphones. Scripts and a ReadMe are on GitHub at **magpi.cc/ghweatherman**.



DARVA: The Event Assistant

This amicable automated assistant was designed to help its makers explain their projects at tech events. **Nicola King** investigates

MAKER

Dane & Nicole

Two makers that create tremendously terrible tech, which they happily share with you on their YouTube channel called '8 Bits and a Byte.

hackster.io/8bitsandabyte

While Brussels-based makers Dane and Nicole enjoy meeting people, they find that endlessly repeating the stories behind their projects at events can be exhausting. "We presented our tech projects at a number of events in the past year, and we noticed we kept repeating ourselves throughout the day, giving the same talks and same explanations over and over again to the people visiting our booth," explains Nicole. "Whilst it's definitely fun to talk to so many people, it's also very tiring and so we thought it would be great to create an assistant, of sorts, to do this repetitive work for us."

▶ When the screen is touched, the robot responds with various phrases and random movements

Felt fusion

Enter DARVA, a cute little animated robot who just loves to chat, and who runs off a Raspberry Pi 4. Dane says, "DARVA was made by first cutting and sewing all the separate parts of the robot out of felt. Then we took a picture of all these robot parts and cut them out using photo editing software [...]. To bring DARVA to life, we created a webpage (with HTML canvas and JavaScript) to which we added all these photographs and animated them."

Over the period of a week or two, Dane and Nicole's idea really took shape. The most difficult part of the project was the touchscreen orientation: "As the felt robot is standing upright, we thought it would be best to use the screen in portrait mode," says Nicole. "Whilst you can easily change the screen rotation on Raspberry

▲ The robot's sewn felt pieces were photographed and then used to create animations

Quick FACTS

- The project took about 20 hours to complete
- Sewing the felt parts was the most time-consuming element
- Felt is a quick and cheap way to create in a range of shapes/colours
- The animations are run in Chromium
- DARVA is an acronym of 'Digitized Analog Robot Virtual Assistant'

Pi, the touchscreen still worked in landscape mode; the mapping between where you touched the screen and where you clicked was wrong." After a lot of trial and error, they decided to keep Raspberry Pi running in landscape and just rotated all the animations.

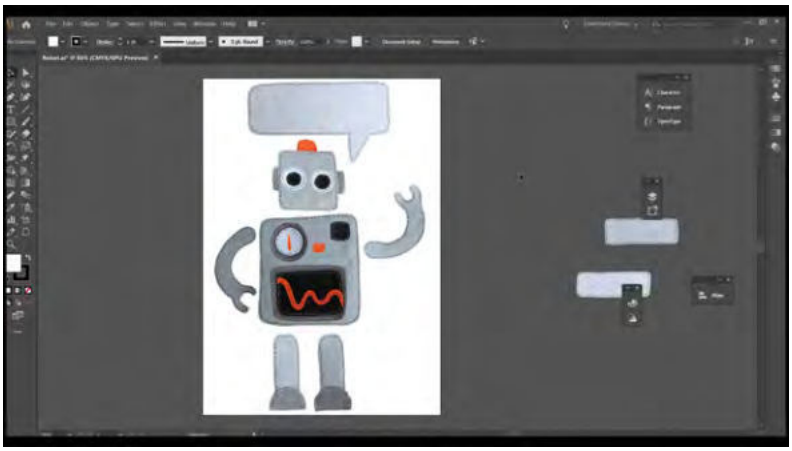
> "They have worked hard to ensure that DARVA has a lifelike quality"

Digital deputy

They have also worked hard to ensure that DARVA has a lifelike quality, and introduced a degree of randomness to the robot's actions. "The gauge rotates to a random position, for example, and the eyes randomly look left or right for a random amount of time," says Nicole. DARVA also loops through a series of texts that Dane and Nicole have written to give some more information about their booth at events, while some animations are activated by clicking or touching parts of the screen, including DARVA's head and belly.

So, how have people reacted to their new digital sidekick? Dane tells us there was some initial confusion at one event: "We placed the original felt version of DARVA next to the touchscreen, because we thought it would be interesting to show how we went from a felt robot to a digital one. However, a lot of people thought the felt robot and the touchscreen were somehow connected and could interact, which was confusing because moving or touching the felt didn't do anything." However, DARVA has received a lot of compliments: "Kids especially loved the look and feel of it."

Dane and Nicole are considering developing the project further. "Maybe we can make it a video game, or an interactive story, or connect the felt robot to the digital robot as many people expected," says Nicole. "We have so many ideas, we hardly know what we eventually will end up making, but one thing is for sure: it will involve a Raspberry Pi!"



▲ The process of animation begins, after the pieces are cut out digitally using photo editing software

Heart-rate monitor

When tasked with a developing a project during their Raspberry Pi classes, these students took the healthy option, as **David Crookes** explains

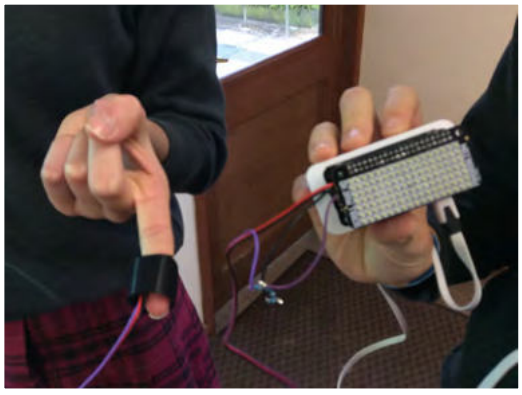
MAKER

Jon Witts

Jon Witts is a certified Raspberry Pi certified educator and the director of digital strategy at Queen Margaret's School in York, where he teaches the Computer Science courses.

@QM_CompSci

Our hearts often feel like they're skipping a beat when we see the latest ingenious projects from the ever-talented Raspberry Pi community. With this project, however, we could go as far as viewing the actual effects they have on our ticker, since a trio of students at Queen Margaret's School in York have worked on a heart-rate monitor using pHATs from Pimoroni and a pulse sensor.

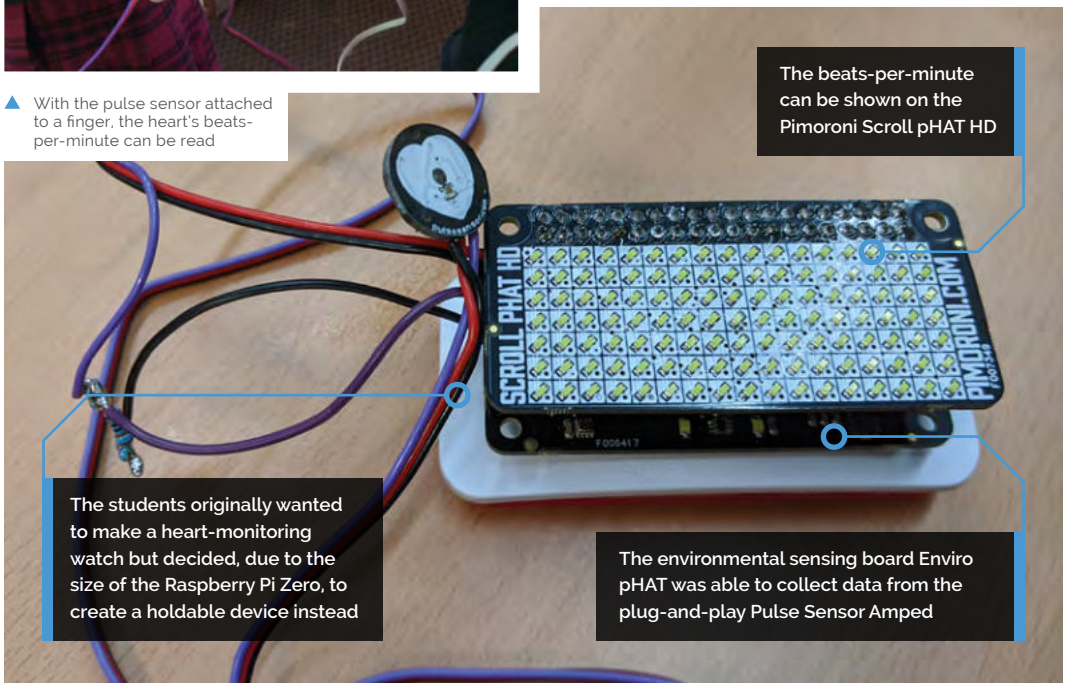


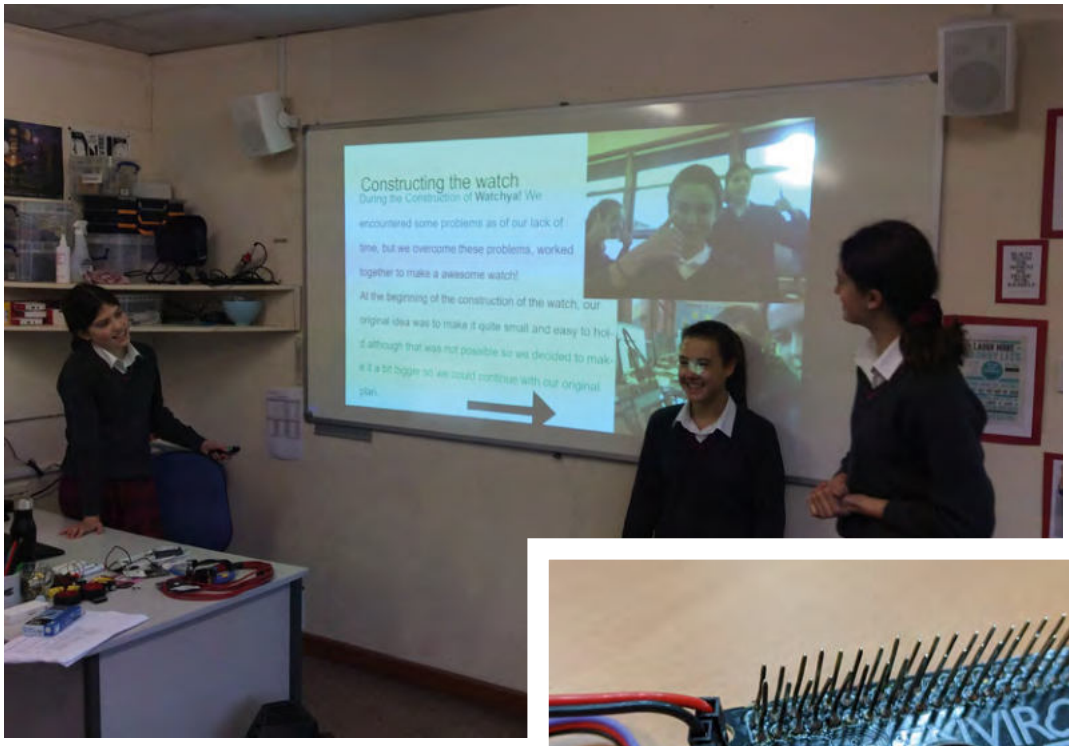
▲ With the pulse sensor attached to a finger, the heart's beats-per-minute can be read

Elena Lardies Lopez, Harriette Pemberton, and Myrtle Morley created the device as part of their computing lessons. "The school teaches a unit called 'Physical Computing with the Raspberry Pi' module and the students learn about connecting electronic devices [and controlling them from] a Raspberry Pi computer before forming small groups and deciding on a physical project," explains Jon Witts, the school's director of digital strategy.

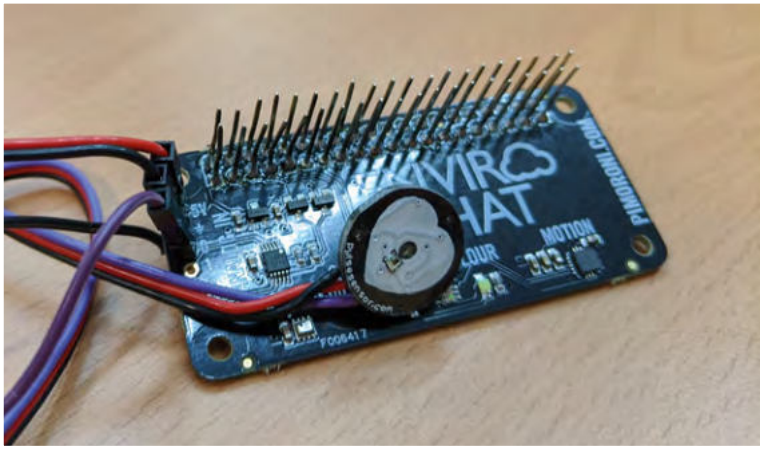
Revised plans

After deciding on their initial aims – to create a watch that could track and display the wearer's step count and heart rate – the students began to plan. They had to break down the problem and research how they could make their project: from the programming involved, to how it would be physically developed, and the hardware that would be needed.





Elena Lardies Lopez, Harriette Pemberton, and Myrtle Morley showcase their class project in a PowerPoint presentation



Two of the main components in this simple build that was used to help students learn more about connecting and controlling devices with Raspberry Pi

"When the students saw the size of the Raspberry Pi Zero they would be using and the size of the pHATs involved, they quickly ruled out having the device as a watch," Jon says. "Instead, the group focused on how they could combine two pHATs (the Enviro pHAT and the Scroll pHAT HD) to record data from an analogue pulse sensor and accelerometer data from the on-board sensors of the Enviro pHAT. The Scroll pHAT HD would be used to display the data recorded."

> The students learn about connecting electronic devices and controlling them from a Raspberry Pi

Jon says the pulse sensor – which fits over the forefinger and allows heartbeats to be measured – uses a simple three-resistor voltage divider to drop its output voltage to 3V. This is then fed into one of the analogue pins on the Enviro pHAT.

Stepping up

As for the steps counter, the accelerometer data is taken from the Enviro pHAT's on-board sensors. "These two data sources are then read into the students' Python program and they're displayed across the LEDs of the Scroll pHAT HD," Jon adds. He says the design makes use of the fact that both the Enviro pHAT and Scroll pHAT HD are I²C devices with different registers for each pHAT. "The students used a header with extra-long pins to solder to the Enviro pHAT; this meant that the Scroll pHAT HD could then be 'stacked' on top of the Enviro pHAT, effectively chaining up multiple I²C devices."

Given more time, Jon thinks the students could strive for greater accuracy: "The project could look at how to take the raw data from the accelerometer and accurately count steps with it, and also how to convert the high and low voltages recorded from the pulse sensor into a count of heartbeats per minute."

But it has taught valuable lessons in project originality and creativity along with teamwork, problem-solving, and presentation skills. "Running this unit has proven that the Raspberry Pi is a great device for teaching physical computing and the students take to it with enthusiasm," Jon says.

Quick FACTS

- A Raspberry Pi Zero is at its heart
- The project cost less than £100
- It displays to a Scroll pHAT HD
- The students made use of Python
- Enviro pHAT measures tri-axis motion

Audio radar

Making radar technology accessible to students with help from Raspberry Pi

MAKER

Dewan Pieterse

An electronics engineer whose mechatronic engineering project for university made excellent use of radar.

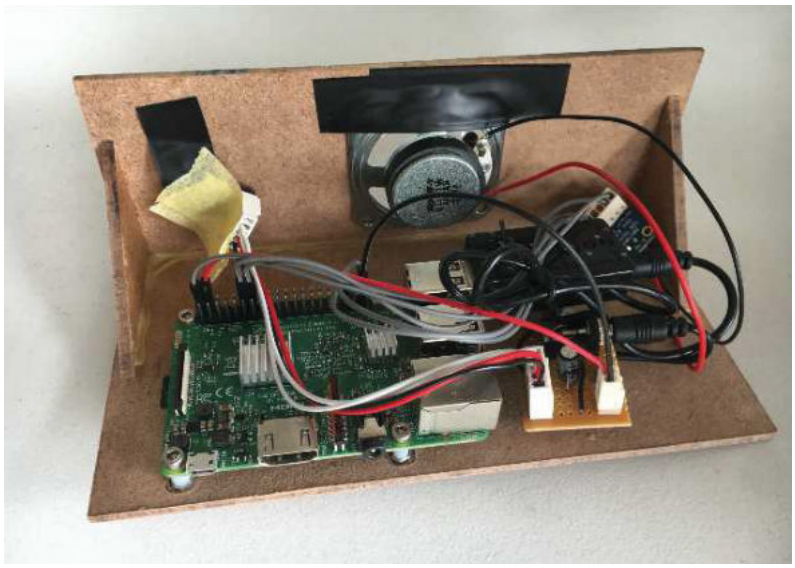
magpi.cc/audioradar

It's well-documented that Raspberry Pi was created to help get people into computing, interesting them in a way that many people were during the 1980s. Dewan Pieterse, formerly of the University of Cape Town, worked on a system to help school and other university students get interested in radar.

"The radar introduces the user in an easy and accessible way to electronics, [as well as] embedded systems and how radar detects return signals and performs filtering to measure radial velocity and distance to objects," he explains.

▶ The radar system can sense distance and speed to varying degrees of accuracy

▼ The innards of the device include a Raspberry Pi, USB sound card, amplifier, and loudspeaker

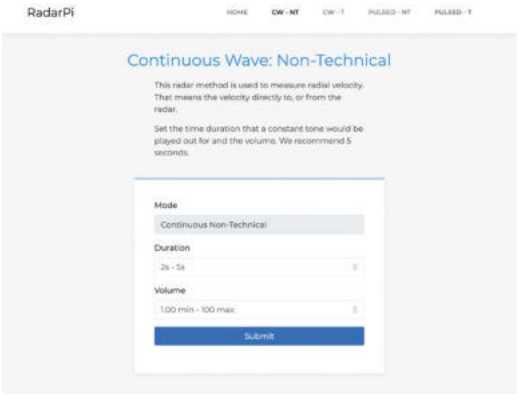


The radar is based on a Raspberry Pi 3B which hosts a web server to enter the parameters needed by the radar to operate. "The web server can be accessed by connecting to the 'RadarPi' WiFi and starts to broadcast upon power-up," says Dewan. "It is also used to display the results obtained after the radar was operated."

What's in a wavelength

Normal radar uses radio waves (it was originally an acronym for RAdio Detection And Ranging); however, this version employs 8 to 12kHz of noise to measure distance – using the reflection time of the sound – and velocity, by making use of the Doppler effect. Parameters can be modified on a webpage generated by Raspberry Pi, allowing students to see how different settings change the results.

"The radar proved to be a minor success in measurement of velocity, [although] with not the desired resolution and clarity due to the automatic gain of the microphone," reveals Dewan. The automatic gain of the microphone automatically adjusts the received signals so that the signal never clips on the voltage range. When the velocity of a car is being measured, the microphone is overwhelmed by the road noise and the resulting plot is a quite faint line representing the velocity of the vehicle.



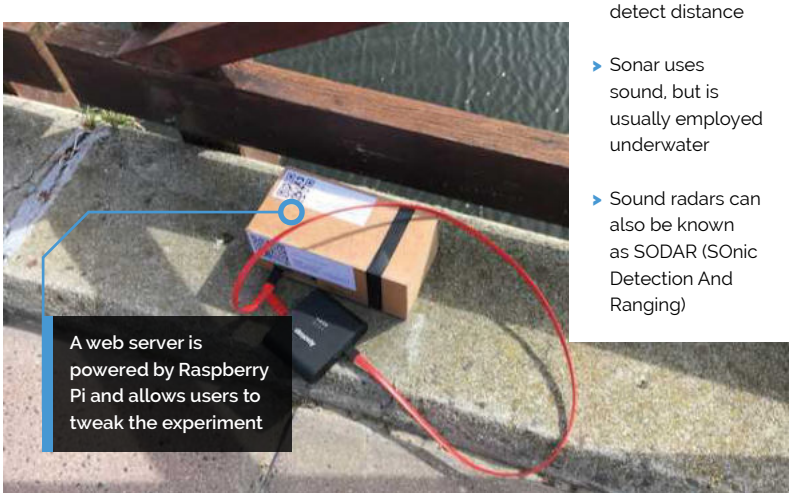
▲ Dewan's RadarPi software enables the user to enter radar parameters via an easy-to-use web interface

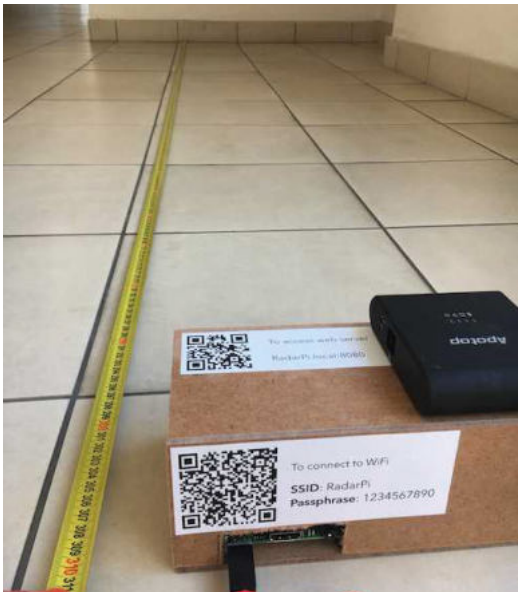
> For distance measurement, the radar proved to be a major success, with the only downside being the resolution of the plot

"For distance measurement, the radar proved to be a major success," he adds, "with the only downside being the resolution of the plot. When an object was measured at a set distance, the results would display the range ±10%. However, the known distance was always in the middle of the range and therefore a successful measurement. The reasons for the variation in distance was not investigated due to a time constraint, but initial tests proved that the automatic gain might be to blame."

Accessible radar

Dewan's system continues a long-running theme of projects involving radar, with this iteration trying to make it as open, affordable, and accessible as possible – which is why a Raspberry Pi was involved: "Raspberry Pi offered built-in wireless LAN and sufficient RAM capacity to host a web server," says Dewan. "Other embedded systems, such as the STM32F04, were also considered, as a lot of previous work was done on this and the on-board analogue-to-digital (ADC) was also attractive. The STM and other Arduino-based systems had insufficient storage and RAM capacity, and therefore Raspberry Pi offered a superior solution."

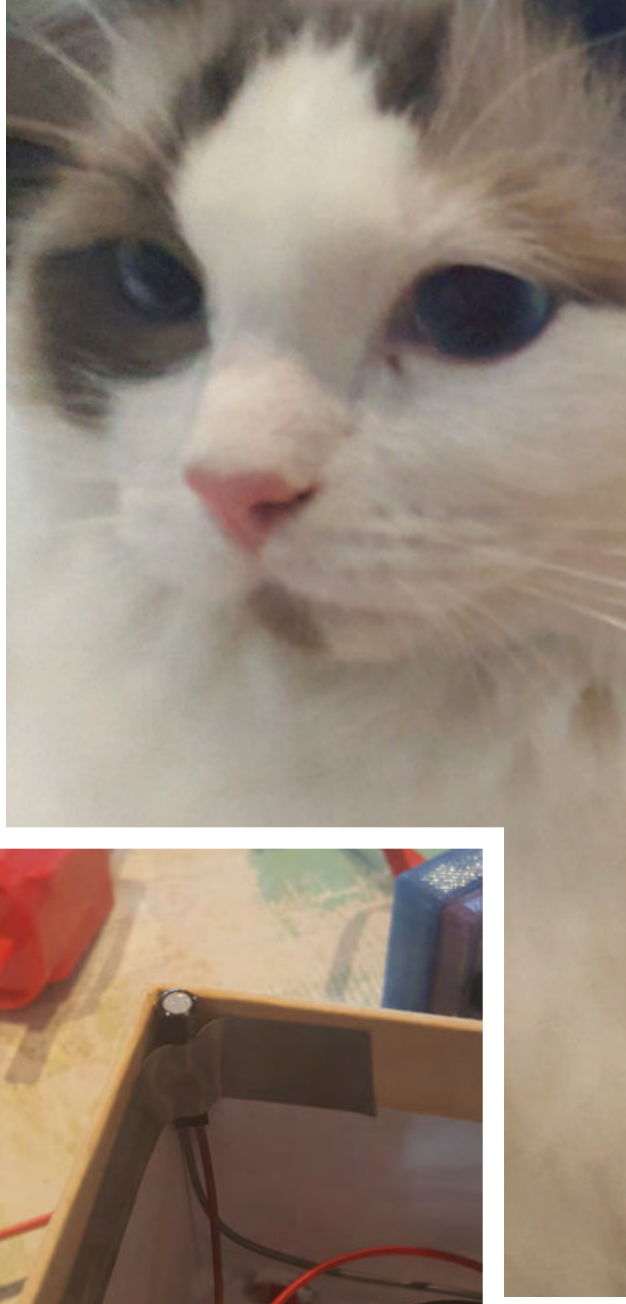




Distance is calculated by measuring the time between the sound being emitted and then received after reflection

Quick FACTS

- QR codes are used to access Raspberry Pi
- Dewan has since graduated from UCT
- Ultrasonic sensors in robots use a similar system to detect distance
- Sonar uses sound, but is usually employed underwater
- Sound radars can also be known as SODAR (SOnic Detection And Ranging)



Tri-Lasers for Felines

Keeping a modern cat entertained requires something more high-tech than a ball of yarn. **Phil King** wonders if this is a purr-fect project

MAKER

Enzo Calogero

Enzo is a field data scientist, passionate about microcontrollers, Raspberry Pi, and 3D printer technologies. He enjoys creating IoT devices to use in day-to-day life, along with robots.

hackster.io/enzo2

Xander the cat is a much-loved family pet, but as his owners live in a flat, he can get a little bored staying indoors when they're out at work. Seeking a way to keep his cat entertained, Enzo Calogero came up with an ingenious Raspberry Pi-powered project. "We noticed that he loves to chase a laser light, so we decide to create a device to make laser games for him," explains Enzo.

The result is the Tri-Lasers for Felines device which, when the cat's presence is detected by a PIR motion sensor, beams a laser dot around the room for Xander to chase between randomly generated points. Judging by the video on the project's Hackster tutorial page (**magpi.cc/trilasers**), he seems to love it.

Pan and tilt

The laser's main movement trajectory is handled by mounting it on a Pan-Tilt HAT, which has vertical and horizontal servo motors. "A pair of coordinates (x, y) is generated randomly," explains Enzo. "The laser point moves from the current point to a new coordinate, following the segment that connects the two points, at a speed defined by a status variable. Once the new coordinates are reached, we loop back to point one."

To add extra interest for Xander, its movement is randomised further by switching between three laser diodes to perform micro random movements very quickly. "Switching the active laser among the three allows extremely rapid movements of the laser dot, to create an extra variability of the light trajectories which seems more enjoyable for the cat," says Enzo.

While the laser point is visible in daylight, it shows up better when there's less light: "Xander prefers it when the room is completely dark."

▲ A tilt sensor was added to shut down the device if interfered with by a curious cat

The device's three laser diodes are set into a 3D-printed triangular holder that sits atop the Pan-Tilt HAT's acrylic mount – which would normally be used to hold a Camera Module. Enzo also designed and 3D-printed a case for the PIR sensor.

Cat-a-log

In addition to handling laser movement, the Python script saves a log of Xander's activity: "We check it now and then out for curiosity," says Enzo. "When Xander was a kitten, he was playing with it very often. Now he is a bit older and much more prone to sleep rather than play, we switch it on when we are out in the evening to keep him busy during our prolonged absence."

One issue that came up is that, being naturally curious animals, cats are prone to investigate any new objects. "We try to put it as high and

Warning!
Laser eye!

Don't look into a laser beam, and don't point a laser beam at a somebody's head.
magpi.cc/lasersafety

A PIR motion sensor turns on the laser when triggered by Xander's presence

The device features three laser diodes mounted in a holder on a Pan-Tilt HAT to move it around

Xander is a curious cat who likes to be entertained

> Xander prefers it when the room is completely dark

unreachable as possible, but cats are extremely skilled," says Enzo. "So, he was able to reach the device few times. And the best way to save the device from cat attacks is to make it is as still as possible, so the cat loses interest."

Therefore a tilt sensor was added to the device, to cause it to shut down if triggered by an inquisitive Xander, thus reducing the risk of damage.

This isn't the only feline-focused project from Enzo, who has also built an IoT food scale to monitor when and how much Xander eats, sending the data to a Google Cloud online dashboard. He's now working on a wheeled robot to track the cat with a camera and perform a few interactions – we wonder what Xander will make of that.

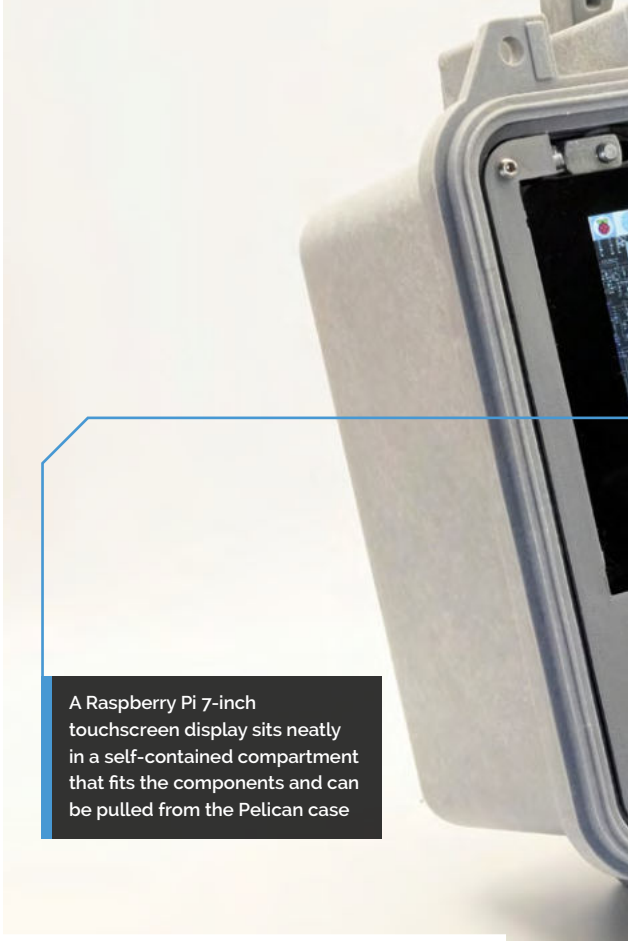
Three laser diodes are set into a 3D-printed holder and mounted on a Pan-Tilt HAT to move the laser dot between random points

Quick FACTS

- The project took a couple of months to complete
- A Pan-Tilt HAT handles the main laser trajectory
- Three lasers are used to produce micro movements
- The software is on GitHub: **magpi.cc/ghtrilasers**
- 3D print designs are downloadable from **magpi.cc/trilasers**

Raspberry Pi Recovery Kit

Jay Doscher's tough cyberdeck could see you through the most testing of times. **David Crookes** logs on



A Raspberry Pi 7-inch touchscreen display sits neatly in a self-contained compartment that fits the components and can be pulled from the Pelican case

MAKER

Jay Doscher

Jay is an IT professional and technology enthusiast based in the San Diego area. He likes to focus on open-source projects. He's on Twitter (**@back7co**) and Instagram (**@back7.co**).

back7.co

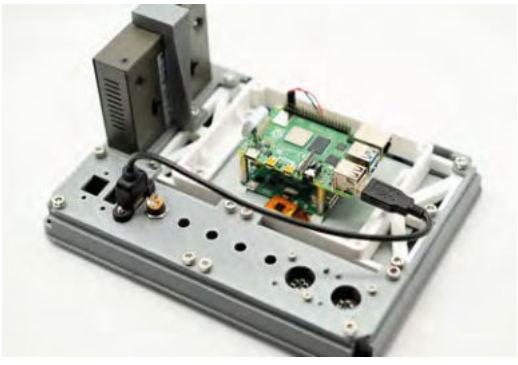
Should the world ever be plunged into an apocalypse, then Jay Doscher should do just fine. He's created a rugged-looking laptop using a Raspberry Pi 4 computer and placed it within a small, air- and watertight Pelican case. Aimed at getting technology up and running in the event of a disaster, it should see him through the most testing of times.

"Most people prioritise food and shelter in the event of a disaster, but what do you do when those are resolved – how do you get technology working again?" he asks. "The apocalypse is more of a thought exercise for me, but I've certainly created a very useful computer that is much easier to work on or modify than a regular laptop."

Past lessons

Jay has been here before. In 2015, he popped a Raspberry Pi 2 into a weather-resistant enclosure and created the Raspberry Pi Field Unit that could run off a 12 V or higher power source, in this case a solar panel. Perfect for outdoor use, it also utilised an Adafruit real-time clock to retain accuracy when off the network. But it was far from perfect.

"The Raspberry Pi Recovery Kit is an evolution of that previous build," he says. "Although each has different goals, I wanted a Raspberry Pi setup that could be rugged and work in a more hostile environment. I also wanted a system that could serve more than one purpose, since the Raspberry Pi platform is so flexible."



The Netgear switch to the left is securely held in place, while Raspberry Pi is mounted on the rear of the touchscreen

The project uses a compact, smart ortholinear keyboard – that is, one which aligns every key to a strict grid rather than have a staggered key layout

One of the first issues he looked to address was the original lack of a keyboard. This time around, he bought a Plaid keyboard kit and, to his delight, noted that it was a perfect fit for his Pelican 1300 case. Jay also chose to use the official 7-inch Raspberry Pi touchscreen. This did away with the need for a mouse, while freeing up a much-needed USB port on Raspberry Pi 4.

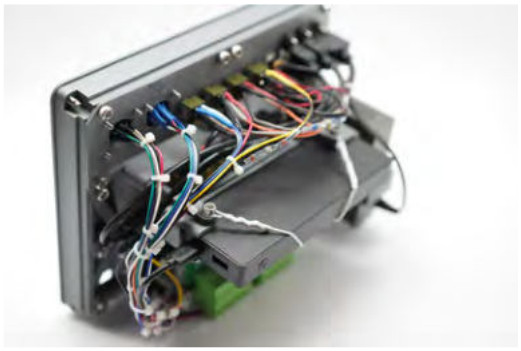
With attention paid to tight wiring and realising that he could get away with powering the unit using 5 V, thereby reducing the need for 12 V circuitry, it wasn't long before the project began coming together. "A Raspberry Pi computer is perfect for this project because it's small, flexible on GPIO, and

Five Ethernet ports run down the right side of the main unit. Vents above the connector panel and display keep air circulating

This smart connector panel can link to an external power supply, give access to the GPIO pins, and connect the keyboard. Switches can isolate power

Quick FACTS

- No holes were drilled into the Pelican case
- The entire device is kept watertight
- The main frame took 24 hours to 3D-print
- It can be powered internally and externally
- It's stored in an electromagnetic pulse shielding box



Here you can see how the connector panel is wired from the back and the positioning of the internal battery

> It's a regular Raspberry Pi laptop in many ways

has great support for third-party add-ons like the GPIO breakout HAT I used," Jay says.

Future-proofing

For a neat interior and to ensure all of the components could be easily held in place, internal parts were printed on a Prusa i3 MK3S 3D printer. For the host of connectors, a panel was produced with locking switches that could turn individual components on and off. These allowed control over Raspberry Pi 4, display, and Netgear five-port Ethernet network, saving power in a potential emergency. A switch also allows toggling between an internal and external battery.

"The internal battery has been the most difficult part, and I am still working on that," Jay says. "There are no real considerations on the Raspberry Pi board itself for battery management, and Raspberry Pi 4 was pretty power-hungry when I built this kit."

Thankfully, coding proved easier. "It's a regular Raspberry Pi laptop in many ways, but I am working on scripts to mirror my GitHub projects, Wikipedia, and Raspbian APT libraries while following their guidelines on proper mirroring," he reveals.

The result is a cyberdeck that can work as a portable standalone network core if needed. "It's a great system to keep air-gapped from the rest of the network when not in use," Jay concludes. The battle for survival starts here.

SUBSCRIBE TODAY FROM ONLY £5

SAVE UP TO 35%

Subscriber Benefits

- **FREE Delivery**
  Get it fast and for FREE
- **Exclusive Offers**
  Great gifts, offers, and discounts
- **Great Savings**
  Save up to 35% compared to stores

Rolling Monthly Subscription

- **Low monthly cost** (from £5)
- **Cancel at any time**
- **Free delivery to your door**
- **Available worldwide**

Subscribe for 12 Months

£55 (UK) £90 (USA & Rest of World)
£80 (EU)

Free Raspberry Pi Zero W Kit with 12 Month upfront subscription only (no Raspberry Pi Zero Kit with Rolling Monthly Subscription)

Subscribe by phone: **01293 312193**

Subscribe online: **magpi.cc/subscribe**

Email: **magpi@subscriptionhelpline.co.uk**

JOIN FOR 12 MONTHS AND GET A

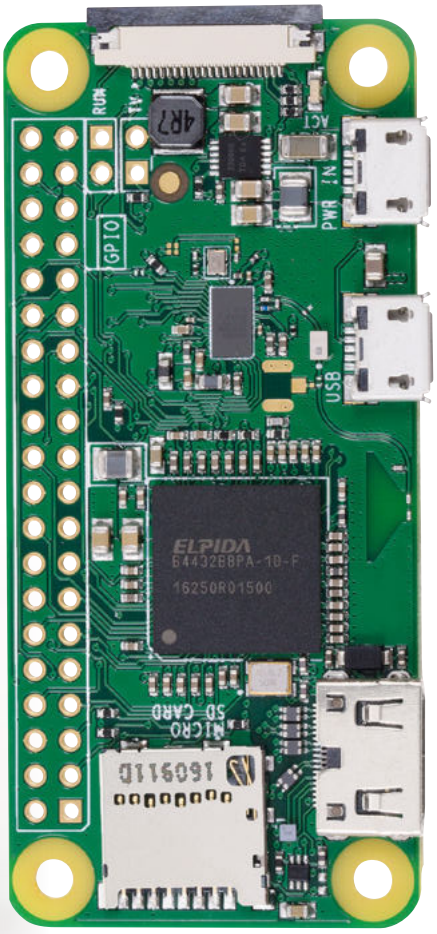
FREE Raspberry Pi Zero W Starter Kit

WITH YOUR SUBSCRIPTION

Subscribe in print for 12 months today and you'll receive:

- Raspberry Pi Zero W
- Raspberry Pi Zero W case with three covers
- USB and HDMI converter cables
- Camera Module connector

Offer subject to change or withdrawal at any time



Buy now: magpi.cc/subscribe

SUBSCRIBE on app stores

From **£2.29**

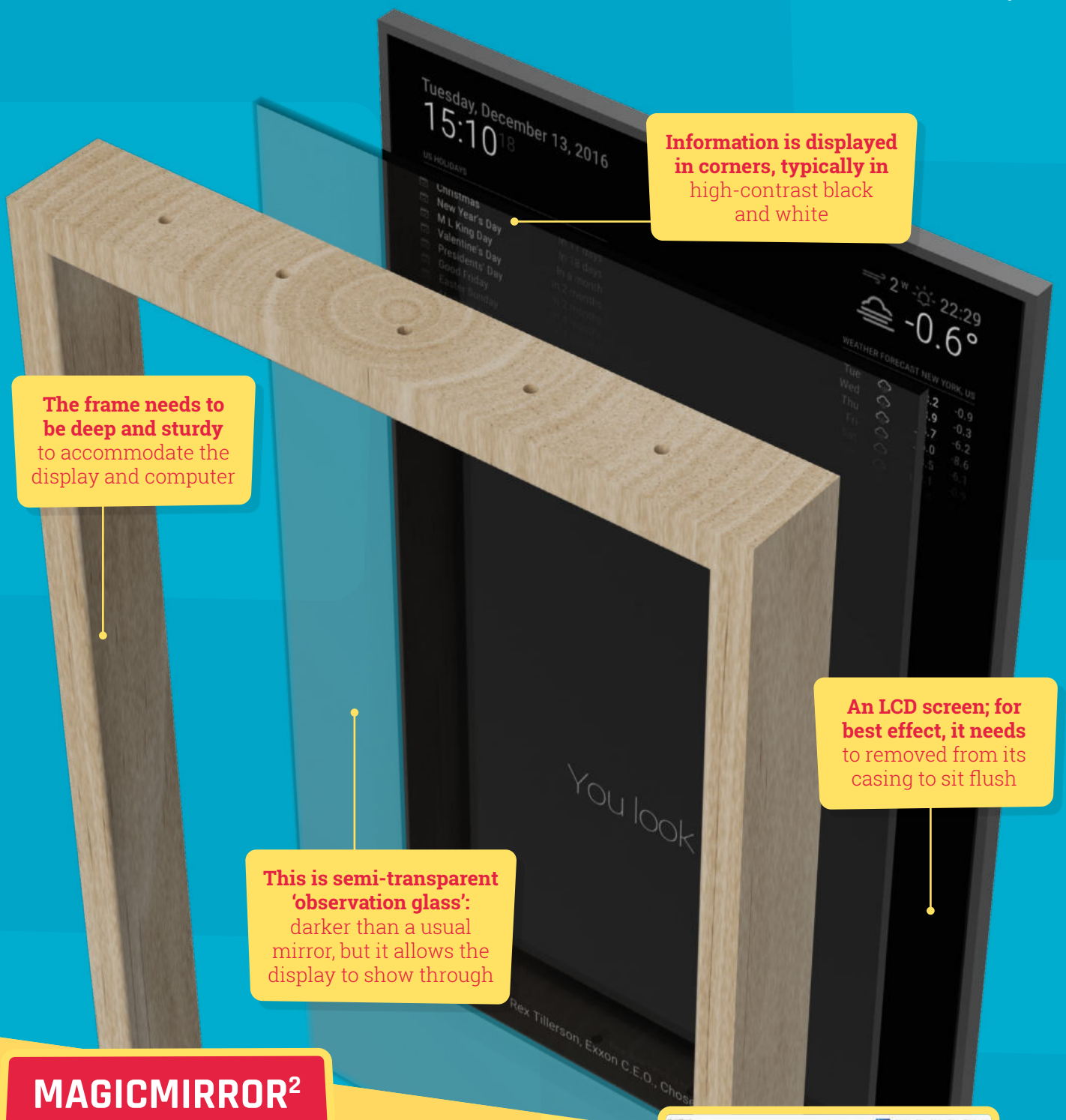
Available on the App Store

GET IT ON Google Play

BUILD THE ULTIMATE MAGIC MIRROR

Magic mirrors. How do they work? Are they really magic? Are they only for the advanced maker? **PJ Evans** goes through the looking glass

Magic mirrors have to be one of the most popular projects out there. Initially created by combining old laptops and semi-reflective observation glass, they appear as normal mirrors but with text and images that appear to float in mid-air. The information displayed is typically what you need as you're preparing to leave the house: weather, news headlines, and transit information. Although they come across as advanced builds, the community behind the projects have made significant advances in making magic mirrors accessible to all. Let's take a look at one of these makers and then have a go at building our own mirror.



Information is displayed in corners, typically in high-contrast black and white

The frame needs to be deep and sturdy to accommodate the display and computer

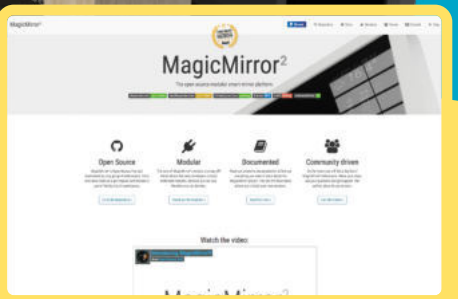
An LCD screen; for best effect, it needs to removed from its casing to sit flush

This is semi-transparent 'observation glass': darker than a usual mirror, but it allows the display to show through

MAGICMIRROR²

Creating a good magic mirror requires experience in many disciplines including carpentry, electronics, programming, and graphic design. Fortunately, the team at MagicMirror², headed by Michael Teeuw (see *The MagPi* issue 54, **magpi.cc/54**), have not only compiled tutorials and fostered a great community, they've also built their own open-source application. This modular system takes away all the programming and design pain. Best of all, you can expand the capabilities of your mirror through the hundreds of community plug-ins available and, if you wish, you can write your own. It's no wonder it won the number one slot in our best projects feature for issue 50.

magicmirror.builders



Assembling a simple magic mirror

Would you like a magic mirror, but don't fancy all that carpentry? Here's a first project to ease you in without having to reach for the band-saw

You'll Need

- Ikea Ribba 18×24 picture frame **magpi.cc/ribba**
- Official Raspberry Pi 7" Touchscreen **magpi.cc/touchscreen**
- A5 observation mirror **magpi.cc/a5mirror**
- Gaffer or duct tape
- Velcro or foam sticky pads

Tip! Not for Zero

A Raspberry Pi Zero would seem ideal for this project, but MagicMirror² is incompatible with that model and the original Raspberry Pi 1.

There have been some impressive magic mirror projects as makers around the globe challenge each other to improve on previous designs. Although the results are undoubtedly impressive, it can make the hobby look a little daunting to the beginner, especially if you don't have access to the necessary equipment to build a custom frame. In this tutorial, we'll assemble a simple magic mirror using off-the-shelf parts. This can be built in an afternoon and is a great way to find out whether you want to take it to the next step and get working on something a bit bigger.

01 Prepare the frame

To create our magic mirror, we will create a 'sandwich' of the frame, a piece of observation mirror acrylic, and the screen. It's vital that all these are kept as clean as possible during assembly as any dust will get trapped and leave an irritating mark on your lovely mirror. Unpack the frame, remove the mount, and then remove the plastic clear sheet. You'll need to carefully peel back the two protective layers and then replace the clear sheet in the frame. This is statically charged and will start to attract dust, so lots of cleaning is required. Return the mount to the frame.

The Ikea Ribba range is perfect for a starter project like this thanks to having unusually deep frames and a wide variety of sizes

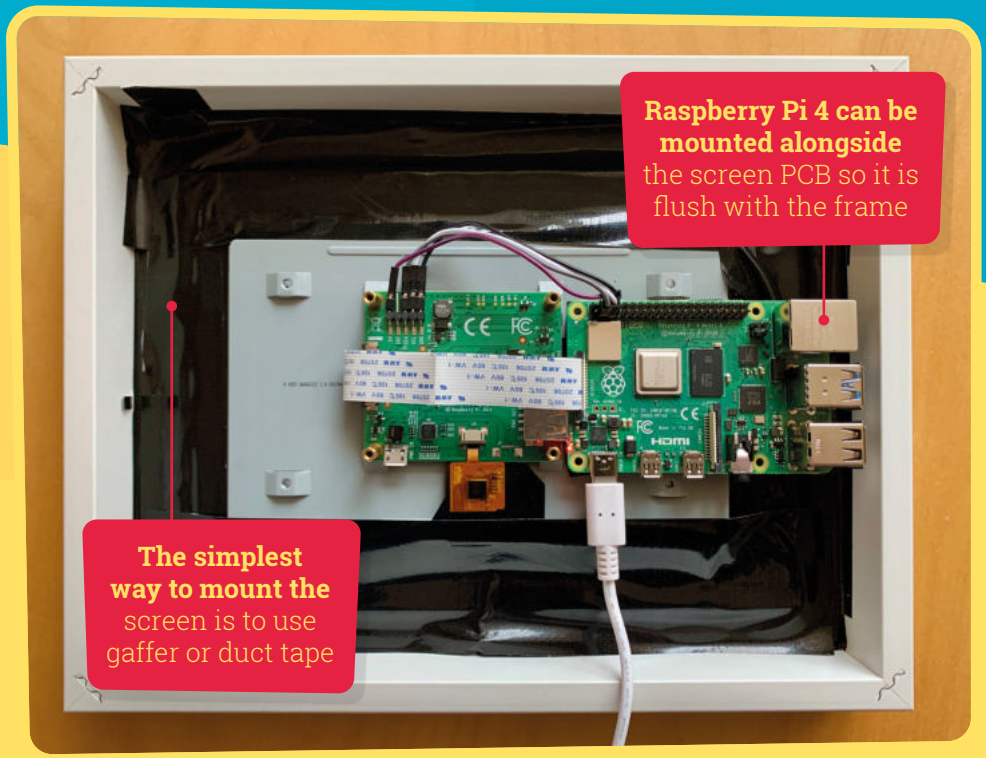
02 Mount the mirror

The big 'trick' of a magic mirror is the use of two-way material, also known as 'observation glass'. This material is the same that is used in police interview rooms and as privacy screening. It's only semi-reflective, so the output from your screen can be seen 'through' the glass but it's still effective as a mirror (if a little darker than a regular mirror). This material is cheapest when bought by the roll, so it's ideal for custom-build or larger mirror projects. Ours is a £5 A5 acrylic sheet. Remove the protective sheeting and place in the frame, making sure it covers the open area. Secure with sticky tape.

> "The big 'trick' of a magic mirror is the use of two-way material, also known as 'observation glass'"

03 Add the screen

We're using the official 7-inch touchscreen for this project to make power requirements easier; we only need one cable to drive both Raspberry Pi and the display. It also happens to be a perfect size for this project. The touchscreen needs to be carefully placed so it's in parallel with the frame and central. Secure with sticky tape.



PROFESSIONAL BUILDS

We've created a simple project for you here that requires no cutting or mains electricity. However, it would be remiss of us not to admire the work of those who have dedicated hours and hours to making the ultimate magic mirror. One of those is MagicMirror² creator Michael Teeuw, who has created several mirrors completely from scratch, building his own frames and carefully mounting large monitors – all powered by Raspberry Pi computers, of course! The great thing about magic mirrors is you can start small and work up to masterpieces like this, learning as you go.

michaelteeuw.nl

04 Secure in place

The combined weight of a Raspberry Pi computer and the touchscreen doesn't come to much, so rather than getting into complicated mounting solutions, we will apply generous amounts of gaffer (or duct) tape to hold everything in place. This is of course a very lo-fi solution – if you want to go for something more refined, you can consider making use of the mounting points on the screen that can be used with horizontal or vertical bars to attach to the inner edge of the frame. Check for any trapped dust or marks in our 'sandwich' before proceeding.

05 Just add Raspberry Pi

Normally, you would mount a Raspberry Pi computer on top of the screen's PCB on the provided standoffs. If you want to mount your completed mirror on the wall, this poses a problem, as the computer now sits quite a way proud of the frame. Your options are: 1) don't care (not advisable), 2) buy a second frame and fix it to the original to double its depth, or 3) mount the Raspberry Pi computer on the side. We've gone with option three and it just fits, even with the supplied display cable. Make sure you line the back of the screen with insulation tape to avoid any electrical shorts and secure in place with a Velcro pad to allow for future access to the microSD card.

Tip!
Larger screen? Better mount

We've chosen gaffer tape, as the official screen is very light. This would be a terrible idea for a 'full-size' monitor.

06 Check and test

With a microSD card with Raspbian installed, mount the Raspberry Pi 4 into place. Check the display ribbon cable hasn't been stretched too much and the four jumper cables that connect the display to the GPIO are in the correct position. You should now be able to boot and see the Raspbian boot sequence through the display. It will probably look disappointingly dull. Don't worry, we'll address that in the next tutorial. If everything is free of dust, secured, and the display is working, shut everything down (you may need to connect a keyboard and mouse to do this).

Setting up your Magic Mirror

Check you're looking good and fill your brain with useful information with your new frame and MagicMirror²

You'll Need

- Magic mirror (see previous tutorial)

Tip!
Editing JSON

MagicMirror²'s configuration file is in JSON (JavaScript Object Notation). It is a powerful but unforgiving format, so be careful when editing it.

So far, we've got a frame built and ready to go. Next, we'll look at how to set up your Raspberry Pi computer to run MagicMirror², a dedicated application for creating magic mirror displays that takes all the hard work out of the software side of this project. MagicMirror² has been authored by Michael Teeuw, who created the amazing mirror in the beginning of this feature. Not only does the software provide a plug-in system to control what's displayed, it also configures your Raspberry Pi computer for optimal performance as a mirror. That said, there are a few things to do, so let's get started.

01 First boot-up

In the final step of the previous tutorial, we checked that Raspberry Pi would boot correctly in its new home. If you haven't done this already, prepare a microSD card with an image of Raspbian or Raspbian Full (not Raspbian Lite, as we need the desktop). If you have access to a second identical Raspberry Pi computer, it might be easier to use that, as the display will be easier to see, then swap the microSD card when you're done. Go through the first-run wizard that helps you configure WiFi and updates all your software to the latest version.

▼ A clearer view of the screen. You can customise all these panels and place them wherever you like

02 Make access easier

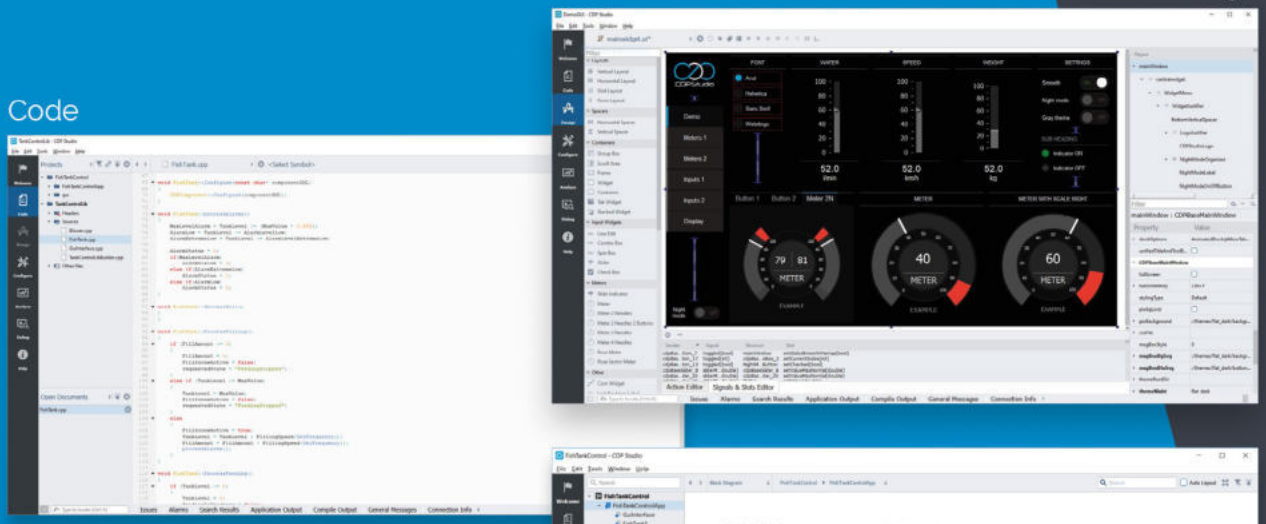
Every time you want to make a configuration change, it's going to be a nuisance to take the mirror down off the wall, locate a keyboard, mouse, and possibly monitor to make a simple edit. Instead we can use VNC and SSH to make life easier. VNC allows us to access the desktop remotely via VNC Viewer, which is available for many different platforms. SSH is similar, but for the command line. To enable both of these, open the desktop menu (the raspberry icon) and click on Preferences, then Raspberry Pi Configuration. When its window appears, click the Interfaces tab and ensure VNC and SSH are enabled, but don't close the window just yet.

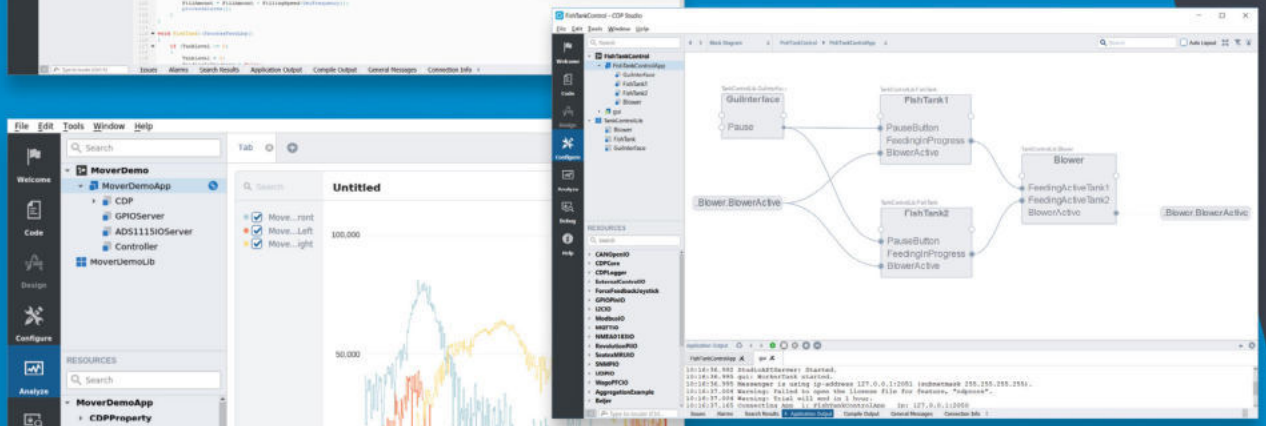
03 Set your host name

To avoid your magic mirror bumping into any other Raspberry Pi computers you may have on your network, we strongly recommend setting the computer's host name (the name it is known as on the network) to something unique. Again, you can set this in the Raspberry Pi Configuration tool. On the System tab, choose an appropriate name such as 'magicmirror' (lower case, no spaces or special characters) and click OK. A reboot will be required for this to take effect. Once done, you can point your VNC Viewer or SSH client to **magicmirror.local** to gain remote access.

Design

Code





Configure

Analyze

NEW GRAPHICAL EDITOR!

CDP STUDIO

Professional control system development tool

CDP Studio is a development platform for industrial control systems, now coming with a free version for non-commercial use. The system can run on a Raspberry Pi, supports C++, open source libraries and has a large feature toolbox including GPIO, I2C and MQTT. Its built in GUI design tool and features lets you code less and do more.

CDP Technologies AS
Nedre Strandgate 29, P.O. Box 144
NO-6001 Ålesund, Norway
Tel: +47 990 80 900 • info@cdptech.com
www.cdpstudio.com

CDPStudio

MagicMirror² defaults to a high-contrast black and white display which works well with the reduced visibility

Your maker. Photographing a magic mirror presents unique challenges

GREAT USES FOR YOUR MAGIC MIRROR

We're going to be looking at uses for your magic mirror in upcoming issues of *The MagPi*. In the meantime, let's whet your appetite with some of the cool things you can do. The link below goes to the exhaustive list of add-ons available for MagicMirror². They cover anything from calendar integration to up-to-the-minute finance information. Need to know how many minutes until the next bus? No problem. Some go beyond the display itself, offering voice control via Alexa and Google Home. How about a Pokémon of the day or video live stream? Can't find what you need? If you know some JavaScript, it is surprisingly easy to write your own modules.

magpi.cc/mirrormodules

Tip! Restarting MagicMirror²

If you have made a configuration change, you can restart without having to reboot by typing this at the command line: `pm2 restart MagicMirror`

04 Install MagicMirror²

Now the fun part. MagicMirror² is an Electron app: a web app wrapped up in its own browser. Installing it manually is complex, but Michael and the team have provided an elegant script that automates the entire process. Get to a command-line prompt by using Terminal or an SSH client and carefully enter the following command:

```
bash -c "$(curl -sL
https://raw.githubusercontent.com/MichMich/
MagicMirror/master/installers/raspberry.sh)"
```

(For more detailed instructions, see **magpi.cc/magicmirror**.)

The script will now install Node (the programming environment that runs MagicMirror²), Electron, and the app itself.

05 Configuring MagicMirror²

During installation you'll see a lot of text fly by the screen, along with a couple of warnings that can be safely ignored. You will be asked at one point whether you want to install and enable pm2. Process Manager 2 (pm2) is an application that simplifies the running of Node apps (or anything else), starting things at boot or automatically restarting apps when they crash. If you reply 'yes', pm2 will start MagicMirror² immediately on boot. You will also be asked if you want to start the app right away. If you're eager to see it in action, say yes.

> "MagicMirror² is now installed and you should be able to see the display"

06 Initial configuration

MagicMirror² is now installed and you should be able to see the display on your mirror. The installation process has tweaked a lot of settings to improve the display quality and has disabled things such as screen sleeping. You can customise what is displayed on screen and its position by having a look at the configuration file located at **~/MagicMirror/config/config.js**. If you're finding the screen too crowded, you can tweak the layout by changing the value of 'zoom'. We found 0.6 worked well for the Raspberry Pi touchscreen. We'll look into more of these settings next time. For now, go and get a hook and put up your new fancy magic mirror!

Voice Control for your Magic Mirror

Mirror, mirror on the wall, who's the fairest of them all? Whaddaya mean, 'Alexa'?

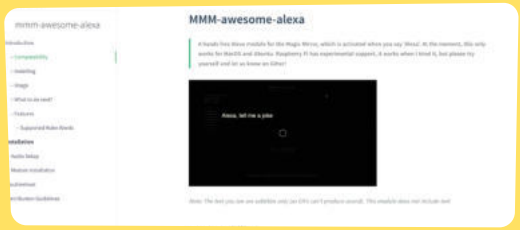
01 Get the right kit

Reliable voice recognition requires a good-quality microphone. If you want to use services such as Alexa, you may want to add an amplified speaker too. The easiest options are widely available USB sound devices, many of which are compatible with Raspberry Pi.

▶ The Snowflake mic is a great choice

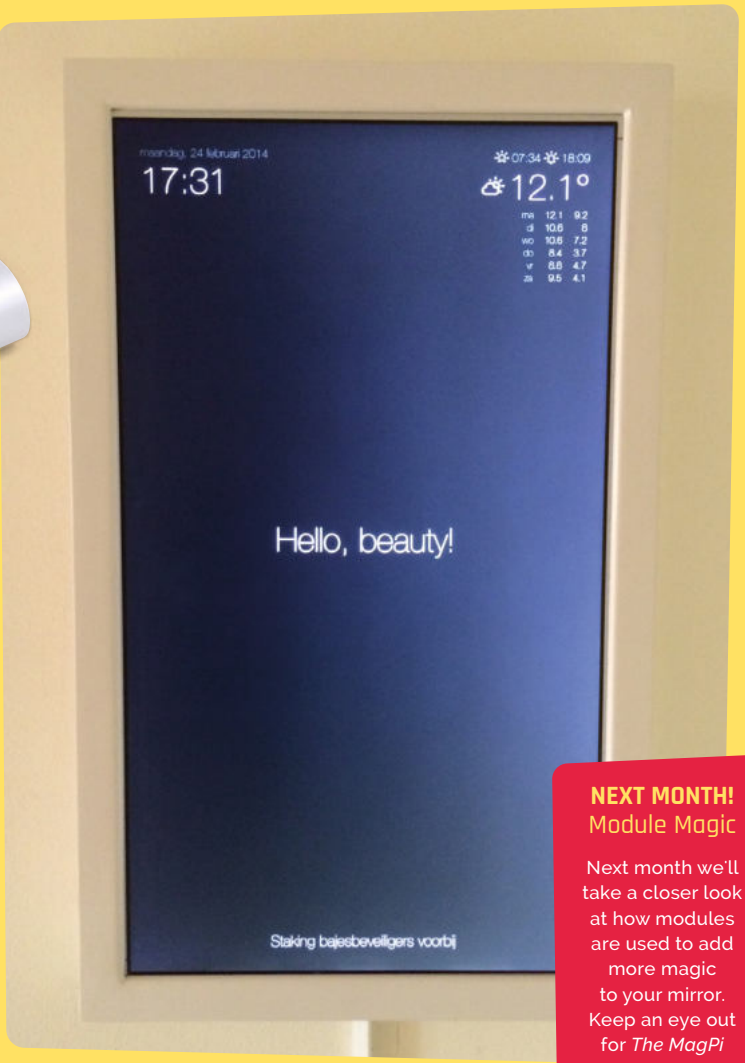
02 Choose your solution

MagicMirror²'s plug-in system and amazing community means you have a wide choice of voice assistants with different purposes and capabilities. See **magpi.cc/mirrorvoice** for the current list of modules.



03 Install your module

A good starting module is one called MMM-awesome-alex. Follow the instructions at **awesome-alexa.js.org** to install the module, then add your microphone and speaker to your mirror build, being careful not to 'muffle' the mic. Now your mirror is Alexa-enabled!



▲ Old monitors and TVs also make great mirror options, and you can still easily add voice control to them!

NEXT MONTH!
Module Magic

Next month we'll take a closer look at how modules are used to add more magic to your mirror. Keep an eye out for *The MagPi* issue 91.


Wireframe
Build Your Own
FIRST-PERSON SHOOTER
in Unity
Learn Unity • Create enemies • Design levels • Make Zombie Panic
Creating a deployable gun turret in Unity
The theory behind first-person hitboxes

Wireframe presents

Raspberry Pi PRESS

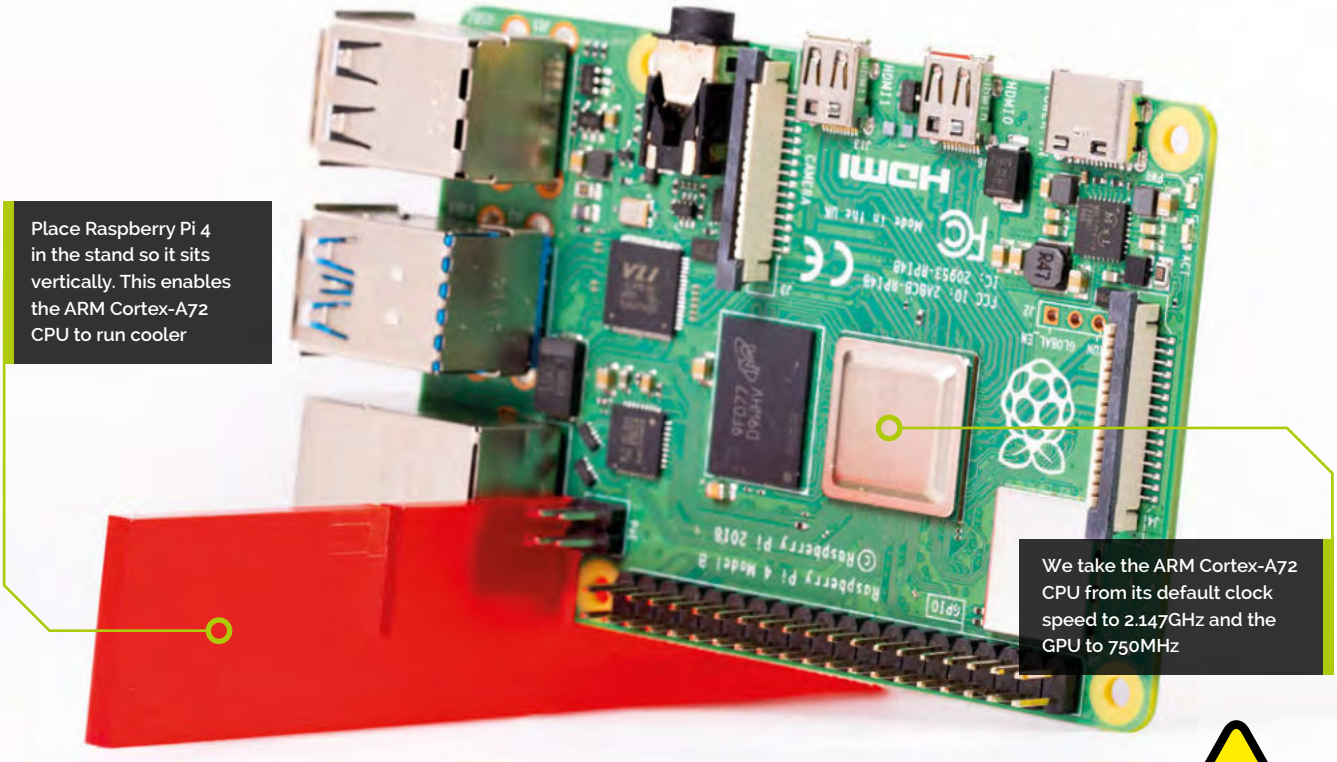
Build Your Own FIRST-PERSON SHOOTER in Unity

Making a fast-paced 3D action game needn't be as daunting as it sounds. *Build Your Own First-Person Shooter in Unity* will take you step-by-step through the process of making *Zombie Panic*: a frenetic battle for survival inside a castle heaving with the undead.

IN THE PROCESS, YOU'LL DISCOVER HOW TO:

- Set up and use the free software you'll need
- Create and texture 3D character models
- Make enemies that follow and attack the player
- Design a level with locked doors and keys
- Extend your game further, with tips from experts

Available now: **wfmag.cc/fps**



Overclock Raspberry Pi 4

Run your Raspberry Pi 4 faster with our guide to overclocking the CPU and GPU. By **Lucy Hattersley**

MAKER

Lucy Hattersley

Lucy is editor of *The MagPi* magazine and enjoys tinkering with gadgets. She's been wanting to write this tutorial for a while now.

magpi.cc

Warning!
Crash likely!

Experimenting to find the highest stable overclock involves crashing Raspberry Pi 4. There is a chance of corrupting the microSD card. Experiment with a clean Raspbian installation and ensure no important data is at risk.

At the heart of Raspberry Pi 4 sits an ARM Cortex-A72 CPU (central processing unit) running at a maximum clock speed of 1,500MHz (or 1.5GHz).

The clock is the electronic pulse used to synchronise all the components in a processor (**foldoc.org/clock**). A maximum clock speed of 1.5GHz means the processor updates one-and-a-half billion times per second.

This is the maximum speed: Raspberry Pi typically idles at 600MHz and switches to the maximum speed when needed. Overclocking is the process of setting a higher maximum speed for computer components. We can adjust the settings in **config.txt** to overclock both the CPU and GPU (graphics processing unit).

We've experimented with speeds up to 2.147GHz for the CPU and 750MHz for the GPU (up from its 500MHz default). These are the kinds of speeds found on high-end desktop computers.

Your mileage will vary, and if Raspberry Pi gets too hot it will slow right down. Experimenting with overclocking will crash Raspbian, and there is a high chance your Raspberry Pi will refuse to start at some point. If programs start crashing, or Raspbian refuses to start, you will need to dial back on the speed. But overclocking is fun and potentially a way to get more from Raspberry Pi.

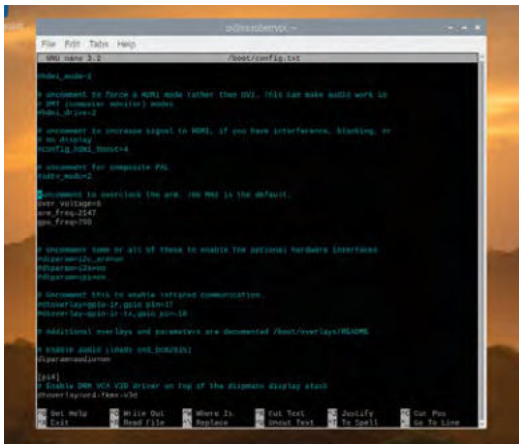
You'll Need

- Raspberry Pi 4
- Raspberry Pi 4 Stand (or cooling case)
- Raspbian OS

01 Raspberry Pi 4 Stand

We start by placing Raspberry Pi 4 in a vertical position. This improves airflow around the components and is surprisingly effective at keeping the temperature down (see Thermal Testing in **magpi.cc/88**).

Use the Raspberry Pi 4 Stand on the front of the print edition of *The MagPi* #90 to run Raspberry Pi in a vertical position. If you don't



▲ Edit the config.txt settings to overclock Raspberry Pi 4

have a Raspberry Pi 4 Stand, you can 3D-print or laser-cut your own with the files on our GitHub page (**magpi.cc/github**).

Alternatively, place your Raspberry Pi inside a case designed to manage the CPU temperature (see our Thermal Cases group test on page 66).

02 Update Raspberry Pi 4

Make sure you are running the latest version of Raspbian OS. Tweaks to performance are being made all the time and you will hit faster speeds with the latest software.

Open a Terminal and enter the following:

```
sudo apt update
sudo apt dist-upgrade
```

Now reboot the system:

```
sudo reboot
```

This restarts Raspbian.

03 Watch your speed

Before we start overclocking, take a look at the default CPU speed. Open a Terminal and enter:

```
cat /sys/devices/system/cpu/cpu0/cpufreq/
scaling_cur_freq
```

Terminal will most likely return `600000`. Divide this result by 1000 and you'll get the speed in MHz. This is the base speed: 600MHz (or 0.6GHz).

Top Tip

Monitoring voltage

It is essential to keep the supply voltage above 4.8V for reliable performance. Note that the voltage from some USB chargers/power supplies can fall as low as 4.2V. This is because they are usually designed to charge a 3.7V LiPo battery, not to supply 5V to a computer.

To monitor Raspberry Pi's PSU voltage, you will need to use a multimeter to measure between the VCC (5V) and GND pins on the GPIO.

More information is available on the Raspberry Pi website (**magpi.cc/powersupply**).

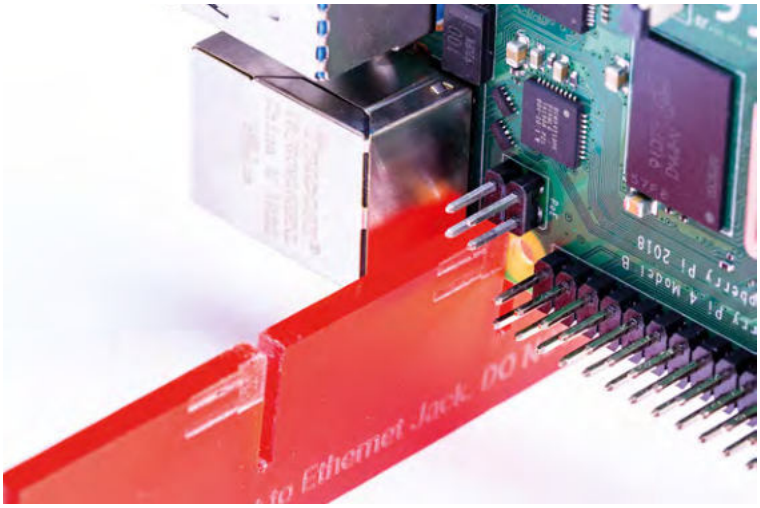
This is the speed requested by the kernel. If your Raspberry Pi is being throttled due to low voltage or over temperature, the actual CPU speed may be lower. To get the actual speed, enter:

```
vcgencmd measure_clock arm
```

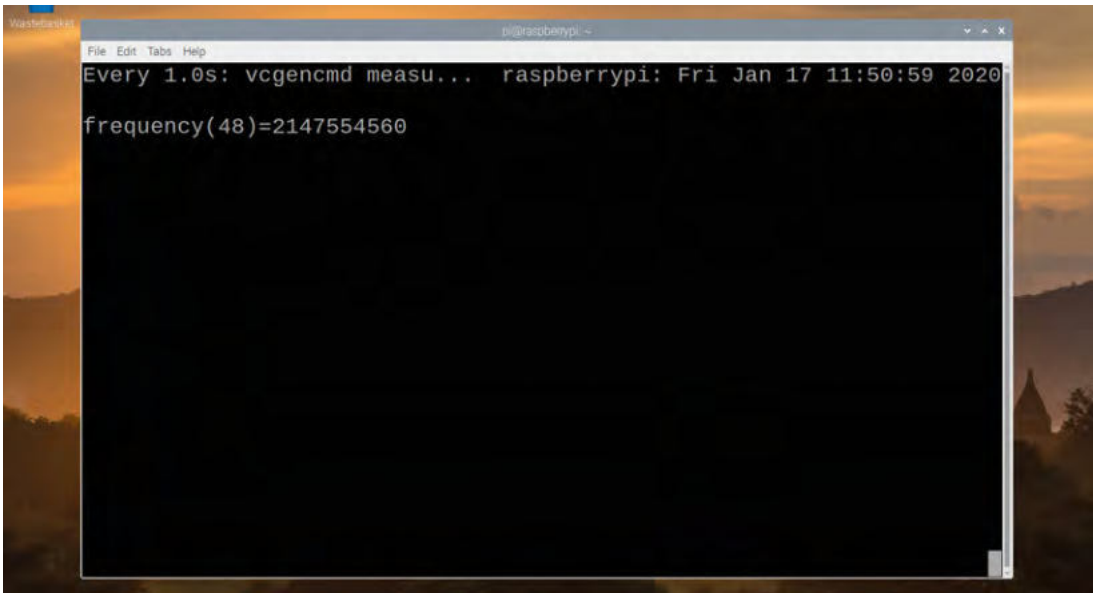
As you use your Raspberry Pi, the requested speed will boost to its upper level, which is `1500000`. You can keep entering `vcgencmd` in Terminal to see where it's currently at, but it's better to use the `watch` command to monitor the speed.

```
watch -n 1 vcgencmd measure_clock arm
```

This keeps vcgencmd running as a process and updates the result once per second (the `-n 1` option is the interval in seconds). Start using your Raspberry Pi and you'll soon see the result go slightly above `1500000` (or 1.5GHz).



▼ Insert Raspberry Pi 4 carefully into the stand, making sure the indented ridge is on the same side as the four PoE pins. This holds Raspberry Pi 4 on its side, enabling airflow around components



The `watch vcgencmd` command is used to keep an eye on the current CPU speed

04 Overclock your config

We're going to use the **config.txt** file to set a new upper limit for the clock frequency. Open another Terminal window and enter:

```
sudo nano /boot/config.txt
```

Scroll down to the section marked:

```
#uncomment to overclock the arm. 700 MHz is the default.
#arm_freq=800
```

And change the settings to:

```
#uncomment to overclock the arm. 700 MHz is the default.
over_voltage=2
arm_freq=1750
```

Save the file with **CTRL+O** (press **RETURN**) and use **CTRL+X** to exit Nano. Restart your Raspberry Pi.

Top Tip

Overclocking problems

Most overclocking issues show up immediately with a failure to boot. If this occurs, hold down the **SHIFT** key during the next boot. This will temporarily disable all overclocking, allowing you to boot successfully and then edit your settings.

Alternatively, remove the microSD card from your Raspberry Pi and insert it into another computer. You will be able to access the **config.txt** file and adjust the settings from there.

```
sudo reboot
```

When the system starts up again, watch vcgendcmd again to see your new, faster clock speed in action:

```
watch -n 1 vcgencmd measure_clock arm
```

Browse a few webpages and you'll see speeds around `1750000000` (or 1.75GHz).

05 Understanding over_voltage

The `over_voltage` command adjusts the core CPU/GPU voltage, and accepts figures between -16 and 8. The default value is 0.

More CPU speed demands higher voltage and if Raspberry Pi doesn't get enough volts, you will see a small lightning bolt appearing in the top right of the window (at this point, the CPU will be reduced to the 700MHz default speed).

06 Crank it up

Let's try taking things a little faster. We're going to take the `over_voltage` setting to 6 and set the ARM CPU to 2.0GHz. Edit the **config.txt** file with the following settings:

```
over_voltage=6
arm_freq=2000
```

This is as high as we're going to take `over_voltage`.

Reboot the Raspberry Pi and you'll be running at 2.0GHz. Run `watch -n 1 vcgencmd measure_clock arm` again to see the new upper limit.

Top Tip

Documentation

See the 'overclocking options in config.txt' page on Raspberry Pi's website for detailed information on all overclocking options. **magpi.cc/overclockingoptions**

07 Take it to the max

Now we're going to boost the `gpu_freq` and take the CPU to its highest setting. This enables Raspberry Pi to run at its current maximum speed. Use Nano to edit the **config.txt** file again, this time setting the `arm_freq` to 2147 and `gpu_freq` to 750:

```
over_voltage=6
arm_freq=2147
gpu_freq=750
```

The `gpu_freq` oversees a range of settings: `core_freq`, `h264_freq`, `isp_freq`, and `v3d_freq`.

The `core_freq` setting adjusts the frequency of the GPU processor. It influences CPU performance because it drives the L2 cache and memory bus.

The default value is 500, and 750 is the highest we can set it and still had a Raspberry Pi 4 run. We have also had Raspberry Pi 4 boards fail to boot at this speed, and others quickly slowed down from overheating or undervoltage. You are unlikely to be able to maintain this speed for the long term and your mileage will vary.

Save the file and exit Nano (**CTRL+O** and **CTRL+X**). Reboot and your Raspberry Pi is hopefully running as fast as it can.

08 Recover from black screen

We have started a Raspberry Pi at speeds of up to 2.147GHz but some of our devices failed to boot, and others showed Undervoltage Warnings (thus reducing the speed). Eventually, we settled for `arm_freq=2000` in config.txt. Our engineering team told us that the benefits from gpu_freq are marginal at best, and it should be removed if Raspberry Pi 4 fails to boot.

Your Raspberry Pi will most likely fail to boot at some point when overclocking. See 'Overclocking problems' for more information on recovery. Otherwise, have fun and we hope you've enjoyed this excursion into overclocking.

Firmware warning icons

Under certain circumstances, the Raspberry Pi firmware will display a warning icon on the display, to indicate an issue. If you see these icons routinely appearing, you should reduce the overclocking speed. There are currently three icons that can be displayed:

Undervoltage warning

If the power supply to the Raspberry Pi drops below 4.63V (+/-5%), this lightning bolt icon is displayed.

Over temperature warning (80–85˚C)

If the temperature of the SoC is between 80˚C and 85˚C, this icon is displayed. The ARM core(s) will be throttled back in an attempt to reduce the core temperature.

Over temperature warning (over 85˚C)

If the temperature of the SoC is over 85˚C, this icon is displayed. The ARM core(s) and the GPU will be throttled back in an attempt to reduce the core temperature.

> We're going to use the config.txt file to set a new upper limit for the clock frequency

config.txt

DOWNLOAD THE FULL CODE: magpi.cc/github

Language: **Bash**

```
001.  #uncomment to overclock the arm. 700 MHz is the default.
002.  over_voltage=6
003.  arm_freq=2000
```

MAKER
Dr Tracy Gardner

Tracy is a computer scientist who spent ten years working as a software architect at IBM. She now writes educational projects for the Raspberry Pi Foundation.

rpf.io/projects

3D-print a keyring with BlocksCAD

Use BlocksCAD to design, and print, a text-based keyring

Last month we introduced our readers to BlocksCAD, a web-based program that can be used with Raspberry Pi to create 3D printable objects. In this project, you will use BlocksCAD to design a 'CODER' keyring that can be 3D-printed.

BlocksCAD is a 3D model editor that you can use in a web browser on a computer or tablet. You drag and drop code blocks to design 3D models that can be exported for 3D printing.

If you have access to a 3D printer, then you can print your keyring. The keyring doesn't use a lot of filament and is small and quick to 3D-print. The keyring will measure around 14 mm by 50 mm. You can put a split ring through the gap between the 'C' and the 'O' to make a keyring.

> "Rendering is the process of creating an image from a 3D model so that we can see what it looks like"

You'll Need

- A 3D printer and filament. Solid colours work best
- A split ring for making the keyring. A 19 mm diameter split ring works well
- BlocksCAD **blockscad3d.com**

01 Create 3D text

Open the BlocksCAD editor in the Chromium web browser on your Raspberry Pi: **blockscad3d.com/editor**. You can drag and drop blocks to write code to create 3D objects (see *The MagPi* 89 for a BlocksCAD primer: **magpi.cc/89**).

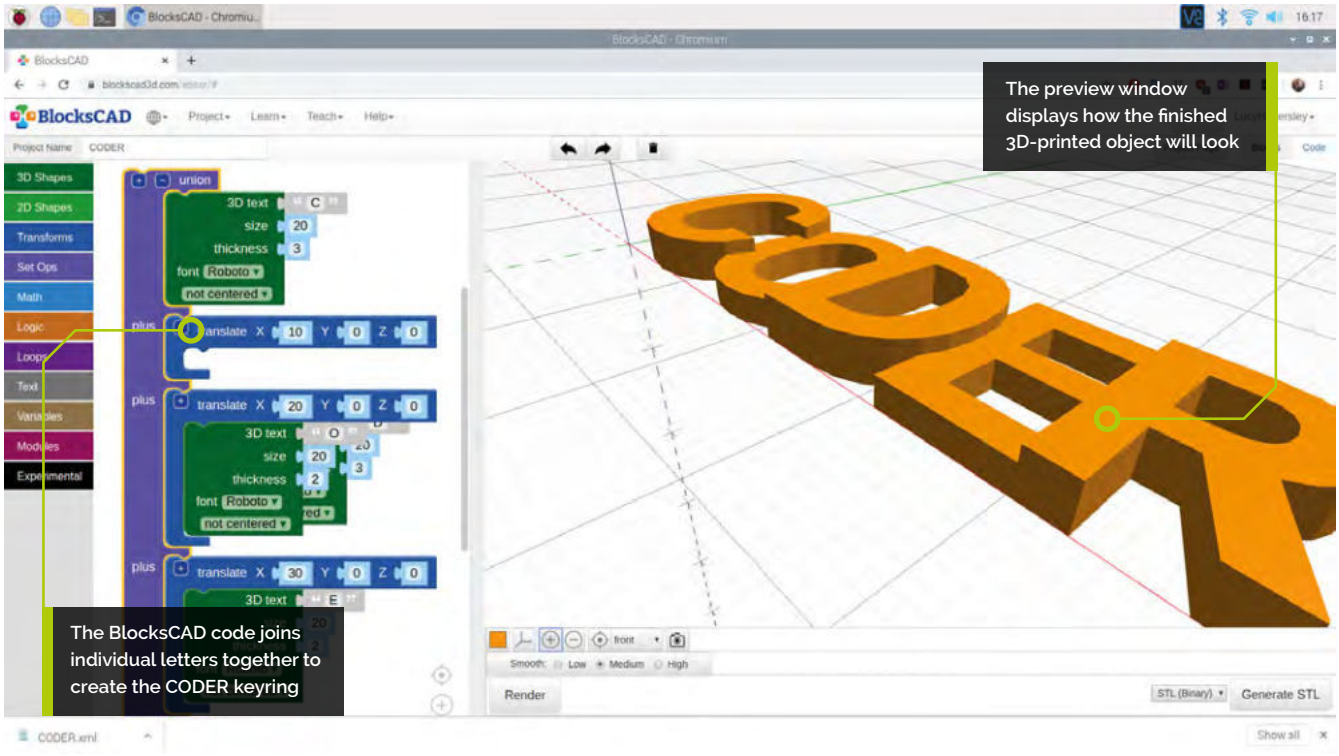
First, use the 3D text tool to create some 3D letters. Click on Text and then drag a **3D text** block onto the canvas. Change the 3D text field to say 'CODER'.

Click the Render button to see the 3D text appear in the preview window. Rendering is the process of creating an image from a 3D model so that we can see what it looks like.

02 Make the text bigger

To make the letters bigger, change the size field to 20 – that means font size 20. Click Render again to see the result. If you 3D-printed the model you have created now, then you would get

▲ Drag a 3D text block onto the canvas



five separate letters, because they are not joined together. That wouldn't make a very good keyring!

03 Join the letters together

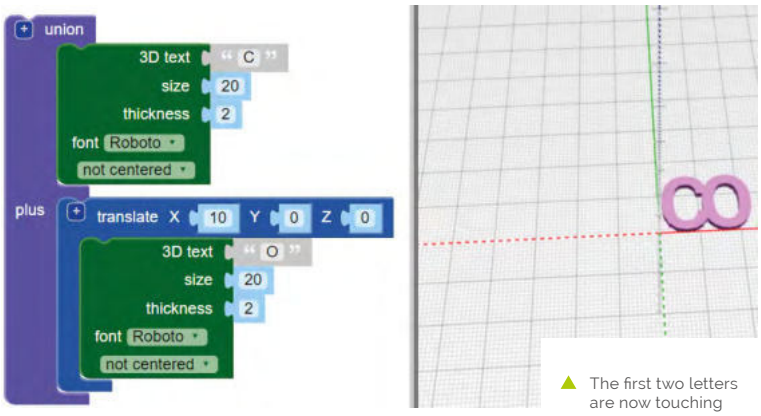
Next, you will join the letters together so that the keyring works as a single object.

Change your existing code so that it only creates the letter 'C'. Add a **union** block, from Set Ops, to join shapes together. Start with the first two letters: 'C' and 'O' (by creating a second **3D text** block and adding both to the **union** block).

Click Render and you'll see that there is a problem: the 'C' and the 'O' are in the same place. You need to move the 'O' along the X axis so that it comes after the 'C'.

Add a **translate** block, from Transforms, and set the value of X to 10 to move the 'O' 10 mm along the X axis.

Now, the letters should be touching, but not on top of each other.



▲ The first two letters are now touching

04 Add the rest of the letters

Click the [+] button on the **union** block to add a space for another block. Right-click on the **translate** block and select Duplicate to create a copy (of it and the **3D text** block inside it).

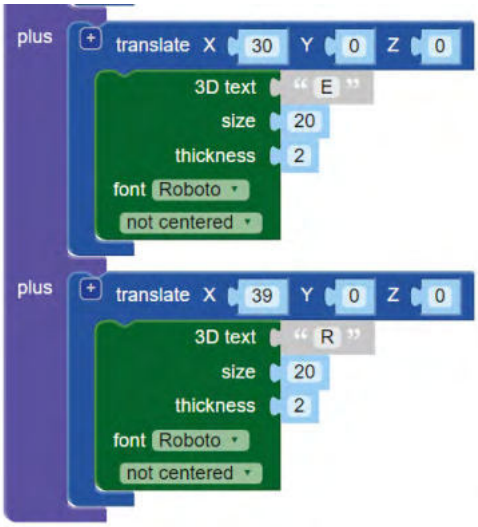
Drag the copy into the **union** block and change the 'O' to a 'D'. Change the X value of the second **translate** block to 20 so that the 'D' is in the right place. Now, follow the same process to add an 'E' and an 'R' to finish the word 'CODER'.

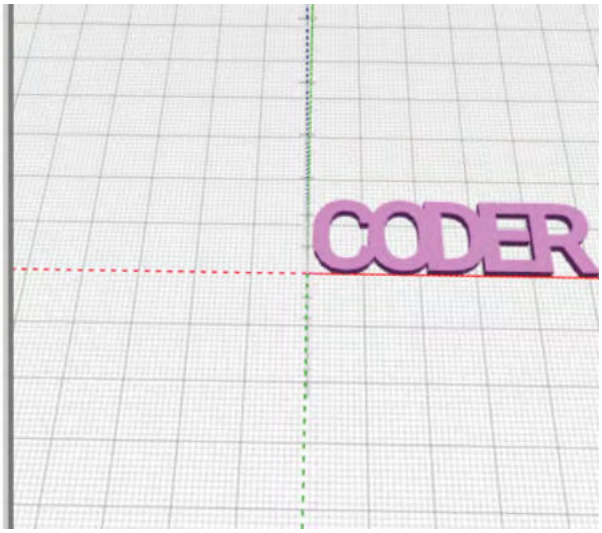
Make sure that all of the letters are touching, and that there are no gaps between any of the letters.

Top Tip

Remove filament

You might need to remove some small strands of filament to tidy up the print.





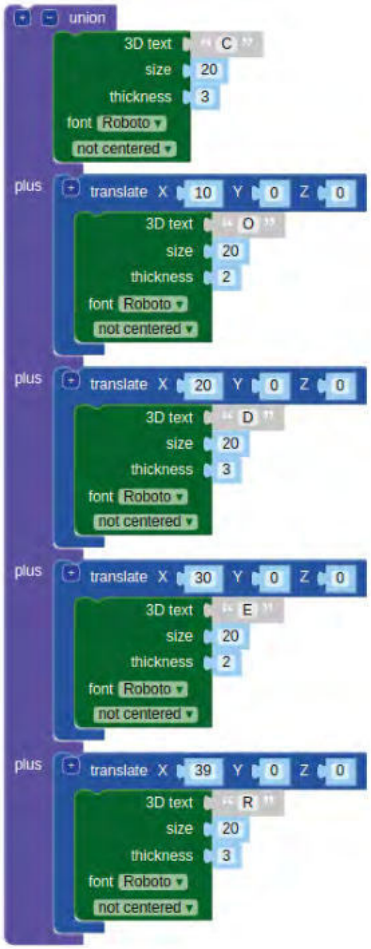
Add the other letters to the union block to spell out 'CODER'

Coder Keyring

Language: **BlocksCAD**

DOWNLOAD THE FULL CODE:

magpi.cc/blockscadkeyring



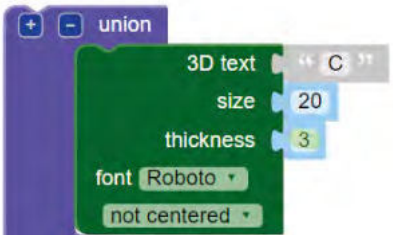
> To save your project, click on Project in the menu bar and select Save Blocks to your Computer

05 Change thickness

Next, we'll alternate the thickness of the letters to help separate them, and to make our keyring even better.

The letters are currently all 2 mm thick; we want the make the 'C', 'D', and 'R' thicker than the 'O' and 'E', by making them 3mm.

To do this, change the thickness value for the 'C', 'D', and 'R' **3D text** blocks to 3, then click Render. Note that you can drag your model around to view it from different angles. You can also click an icon (to the right of the colour) to hide the grid.

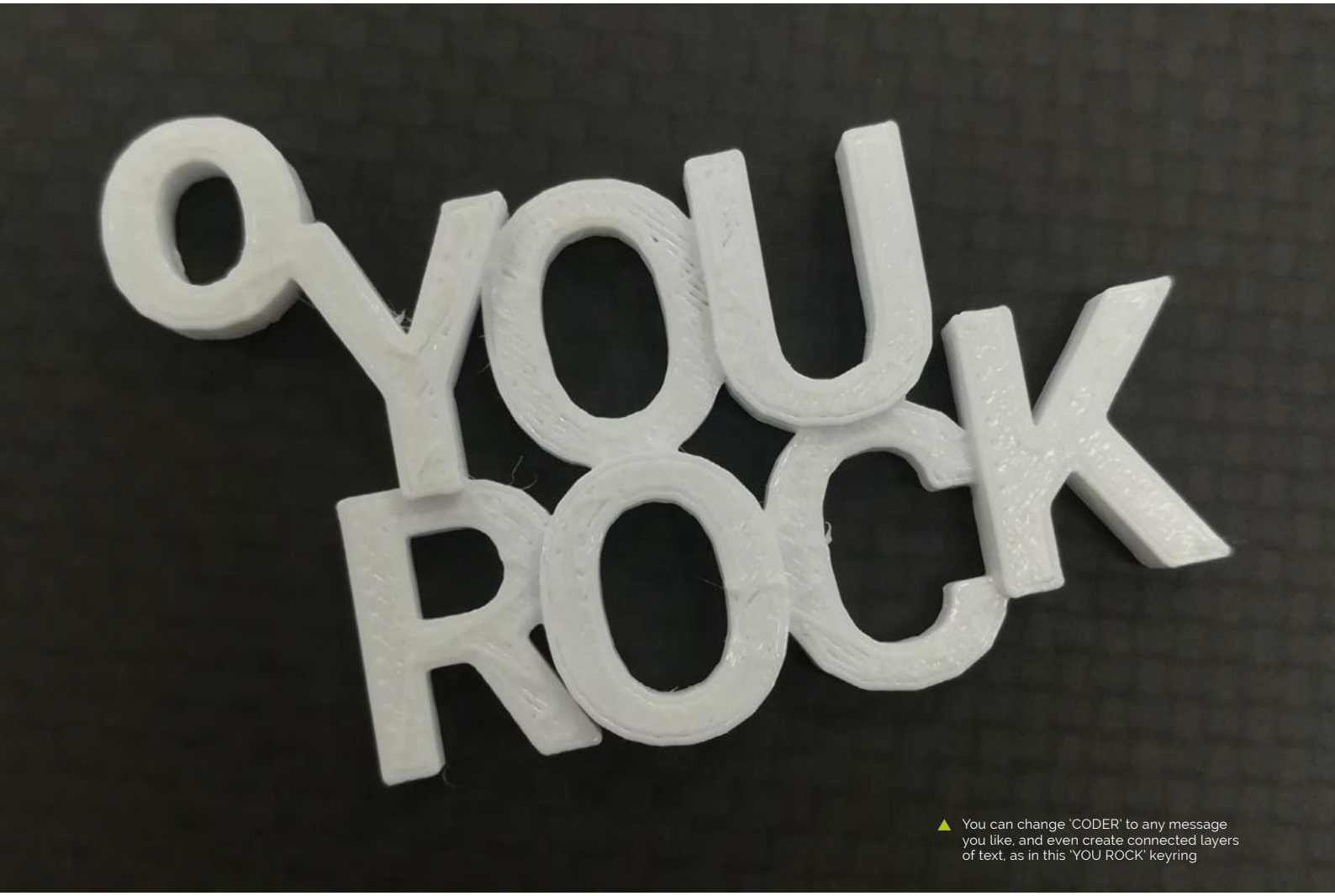


Change the thickness value for three of the letters in the keyring to make it more interesting

06 Save your project

To save your project, click on Project in the menu bar and select Save Blocks to your Computer.

Give your project a recognisable name and remember where you save it. You can then use Load Blocks from your Computer to open the project that you have saved and continue working on it.



▲ You can change 'CODER' to any message you like, and even create connected layers of text, as in this 'YOU ROCK' keyring

If you have a BlocksCAD account and you're logged in, then you can click on Project in the menu bar and select Save. This saves your project online so that you can access it from any computer.

07 3D-print your keyring

BlocksCAD can export an STL file for 3D printing. Render your model and then click Generate STL. Remember where you save the STL file. There are lots of tools that can read STL files and send 3D models to a printer. The one you choose will depend on the 3D printer that you are using. Carefully remove the 3D print from the print bed. Attach the split ring to your 3D printed keyring.

Top Tip

Change colour

You can click on the coloured square to change the colour of your model in the preview window. If you 3D-print the model, the colour of the keyring will depend on the colour of the plastic filament that you use, but it's useful to be able to try out different colours in the preview.

Build an internet-connected room guard

Protect your stuff from nosey parents, annoying siblings, or your nemesis with this easy-to-build motion-detection alarm

MAKER

PJ Evans

A writer, developer, tinkerer, and Milton Keynes Raspberry Jam wrangler. His cat keeps setting the siren off.

@mrpjevans

Warning!
Loud noise!

The siren used in this tutorial is very loud (120 dB). Always use ear protection when testing.

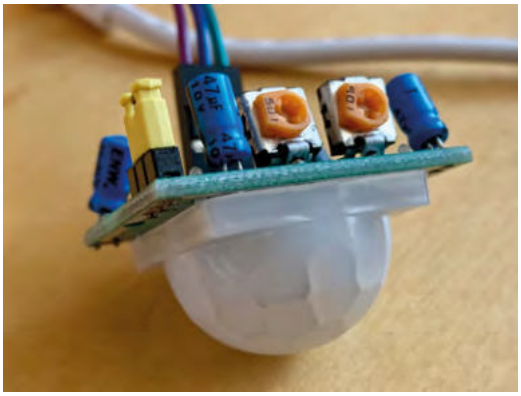
You'll Need

- 12 V indoor siren and PSU **magpi.cc/siren**
- HC-SR501 PIR sensor **magpi.cc/pir**
- Automation HAT **magpi.cc/automationhat**
- Camera Module (optional) **magpi.cc/camera**

If you're just getting into making with Raspberry Pi, this tutorial is a great example of using our favourite little computer for physical computing, reacting to inputs, and doing something useful with the results. It also demonstrates some of the challenges you may face when handling different types of voltage and current. Here you'll learn how to make a simple motion detector that triggers a siren and sends an alert to your smartphone when it's triggered. You can then make the project your own through experimenting and adding more features.

01 Why an Automation HAT?

Many Raspberry Pi physical computing projects directly use the GPIO to connect things like sensors, buzzers, and LEDs. In this case, we're going use the incredibly versatile Pimoroni Automation HAT. This 'input expander' allows us to control devices that would normally either be incompatible or even damage your Raspberry Pi. As a small buzzer may not be particularly deterring to a would-be room invader, we've chosen a 120 dB siren that requires more current that our Raspberry Pi can safely handle. So, we will give the siren its own power supply and safely control it using one of the Automation HAT's relays.



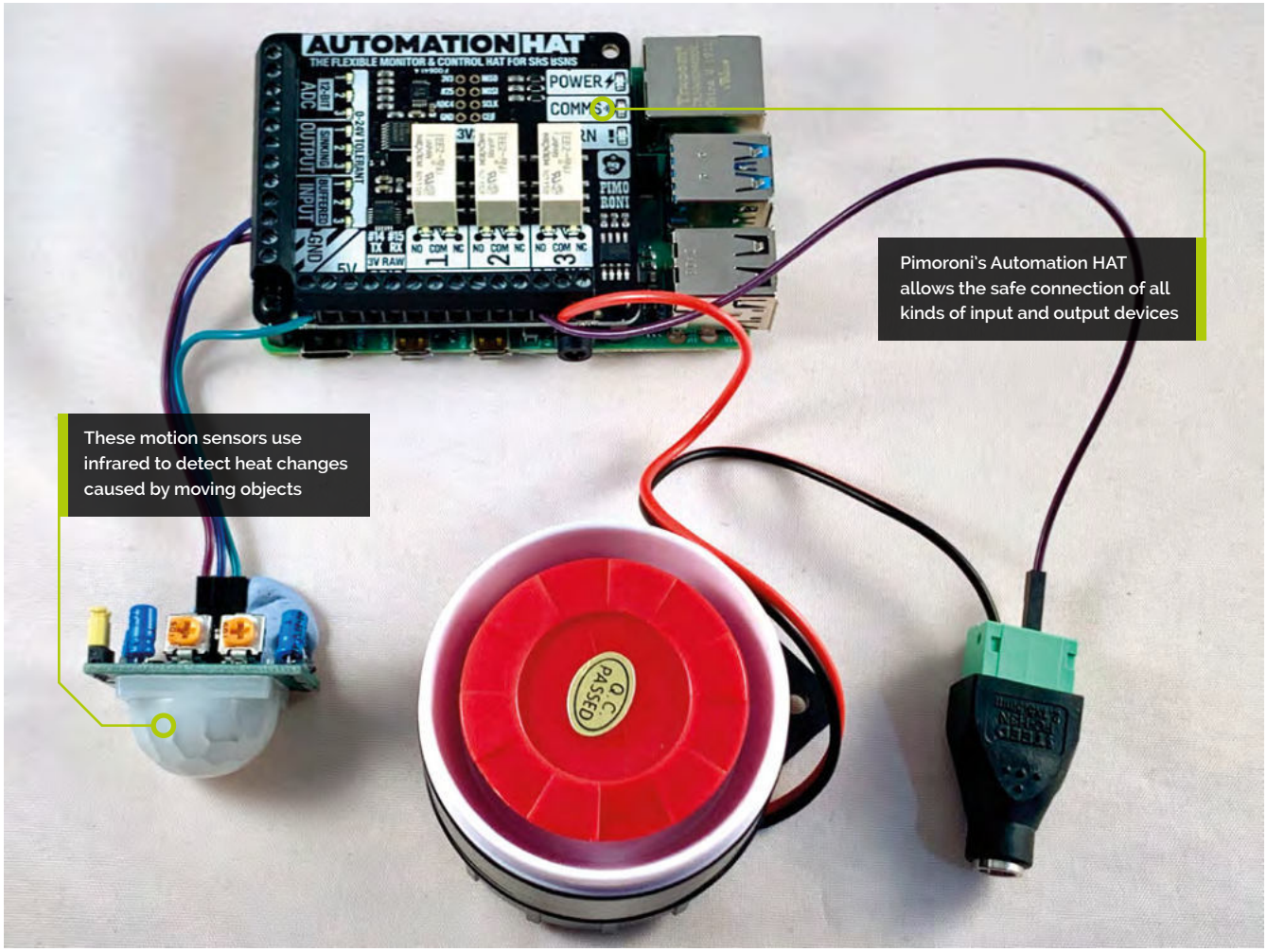
▲ **Figure 1** The PIR sensor's potentiometers (orange). Adjust carefully with a small screwdriver, as they can be fragile

02 Prepare your Raspberry Pi

It's up to you what model of Raspberry Pi you use for this project. A Raspberry Pi Zero W is more than capable of running the code and we originally prototyped our guard using one. If you're thinking of getting clever, though, such as adding facial recognition using a camera, you might want to go for the horsepower of a Raspberry Pi 4. Either way, start by attaching the Automation HAT carefully to the GPIO header, and secure with the provided standoff posts on the opposite side to ensure the HAT doesn't wiggle about and come loose.

03 Set up the software

We're going to use Raspbian Buster Lite as the operating system as there's no need for a user interface, just a basic command-line operating system. That said, there's no harm in installing the full version if you're more comfortable with that. Download the image from **magpi.cc/raspbian** and flash to a microSD card. Now, as ever, it's time to log in and update everything with `sudo apt update && sudo apt upgrade`. Once finished, run `sudo raspi-config` and set up networking (if using WiFi). Now, install the Automation HAT software by running the following command:




```
curl https://get.pimoroni.com/automationhat
| bash
```

This will guide through setting up for driver software and examples. You may need to reboot your Raspberry Pi afterwards.

04 Get sensitive

The PIR sensor has three connectors: one for 5 V power in (which can be provided by the HAT), ground, and in the centre is the data line. Operation is very simple. If the sensor is triggered by movement, the data line goes 'high' (outputs current). After a period of no movement, it goes low (no current). Its sensitivity can be controlled by adjusting one of the potentiometers (the left one when it's turned upside down – see **Figure 1**). The other potentiometer sets the minimum time for which the sensor reports movement. We could connect the PIR directly to the GPIO, but as the HAT is now in the way, we'll use its inputs instead.

Adding alert notifications allows you to keep on eye on things when you're away, and you can add images too

Top Tip

Any buzzer you like

You don't have use a loud siren. If this is just for fun, maybe a buzzer may be better, or could it play a silly MP3 recording?

Top Tip

Calibrate carefully

Be patient when calibrating the sensor. Start with both potentiometers turned fully anti-clockwise, and work from there.

05 Connect the sensor

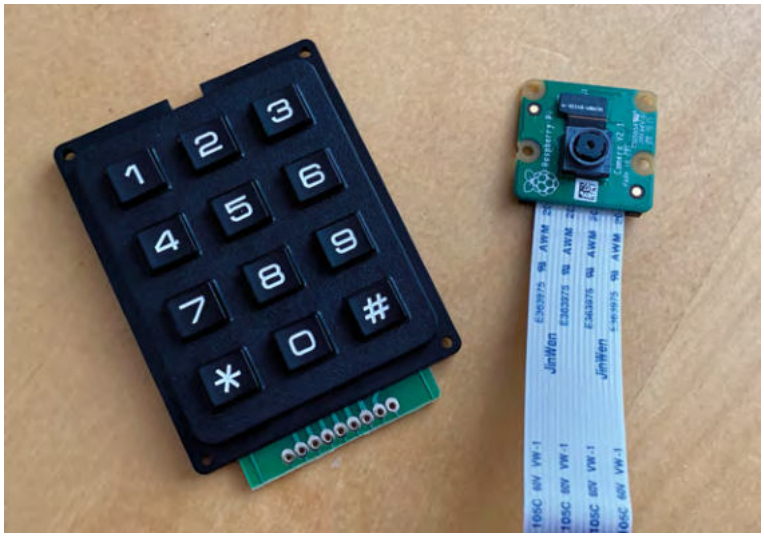
Although this type of PIR sensor is widely available, they do vary in wiring. In particular, the orientation of Vcc (power in) and ground can be different. The sensor we've recommended here has both clearly marked on the PCB. Connect the ground pin to any of the GND connectors on the Automation HAT, then connect any 5 V power output to the 5 V in pin on the sensor. So that we know what state the sensor is in, connect the data line to any one of the buffered inputs on the HAT. Carefully check each of the screw terminals for a solid connection.

06 Time for a test

When you installed the software for the Automation HAT, examples and documentation were included. We can use these to quickly test and calibrate the sensor. From the command line, run the following:

```
python3 ~/Pimoroni/automationhat/examples/input.py
```

You will now be shown the reading from all the input sensors. Watch the one to which you connected the data line in the previous step. It should report a 1 when it is activated (motion has been detected) and 0 when it 'times out'. You can carefully adjust the two potentiometers on the sensor to fine-tune the sensitivity and timings to your preferences. Press **CTRL+C** to stop the Python script when you're done.



There's plenty of scope for adding more to the project, such as an inexpensive keypad or a camera

07 This might get loud

The siren is capable of an ear-splitting 120 dB. Always wear ear defenders in case you accidentally set the siren off. This particular model is tolerant of voltages less than 12 V and the volume reduces accordingly. We recommend starting no higher than 5 V, which is still incredibly loud. The siren requires its own power supply, so you'll need to source one (we recommend something with variable voltages so you can experiment with noise levels). Before continuing, make sure you're happy with the siren's noise levels. If you don't want a loud siren, there are many buzzers and quieter alarms available that will work to demonstrate the project and will be much safer for younger makers to handle.

> "Always wear ear defenders in case you accidentally set the siren off"

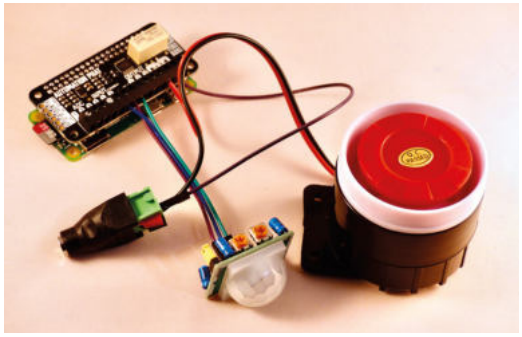
08 Relay race

Relays are magnetic switches, allowing one device to control another without their circuits 'touching'. The Automation HAT comes with three relays and we'll use one to switch the siren on and off. Making sure it's unplugged, snip the siren's positive line (red wire) and strip a bit of wire on both ends. Now, connect the part going to the power supply to one of the relays on the terminal marked 'COM' (common). Connect the other part of the red wire to the same relay on 'NO' (normally open). Double-check everything and make sure the wires are secure.

09 Software

We can now read an input and create an output using the relay. Using your favourite editor, enter the **roomguard.py** code here. This is a simple loop that will check the sensor twice a second to see if movement has been registered. If so, the relay is switched into place, allowing current to flow to the siren and it sounds. Once the sensor no longer registers movement, the relay is deactivated and it all goes quiet. If you don't fancy typing it in, you can download all this code and a few extras from **magpi.cc/roomguard**.

To run the code, enter the following on the command line:



▲ This project works well with Raspberry Pi Zero if you're keeping it simple

```
python3 roomguard.py
```

Try moving about!

10 Add notifications

We now have a working motion-detection alarm, but this is meant to be internet-connected, so let's make it smarter. It would be useful to have a notification sent to a smartphone when the alarm is activated. There are a many different types of notification service, and adding support for most is possible using just one Python library, Apprise. Have a look at the code for **roomguard_notify.py** on GitHub for an example of how to add notifications. Also, check out Apprise's documentation at **magpi.cc/apprise**.

11 Add a camera

Now the basics are working, you can augment the alarm with additional sensors and/or notifications. The Automation HAT has plenty of remaining inputs and outputs for you to play with. A simple step would be to add a Raspberry Pi Camera Module. See if you can change the code to take a photo of the intruder and then forward that image to your notification. Look at the GitHub repository for an example.

12 Make it your own

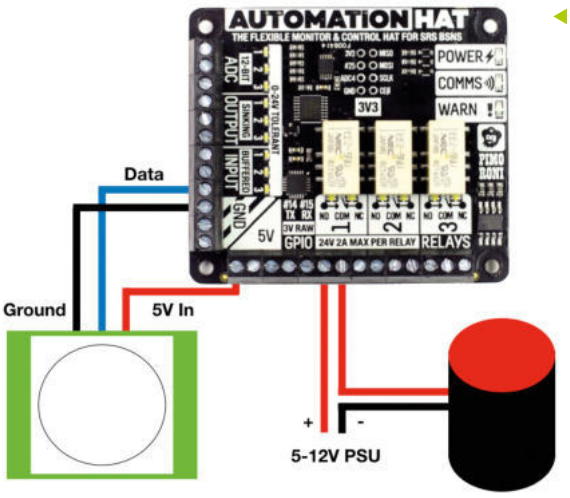
What else could you add? One aspect of an alarm not implemented is any kind of deactivation. Could you add a web server so you can control the alarm from your phone? How about using a keypad to set an activation code? There's potential to add batteries and additional PIR sensors to create a standalone unit. Use facial recognition software to identify who was in your room. There's lots of avenues to explore. Over to you.

roomguard.py

DOWNLOAD THE FULL CODE:
magpi.cc/roomguard

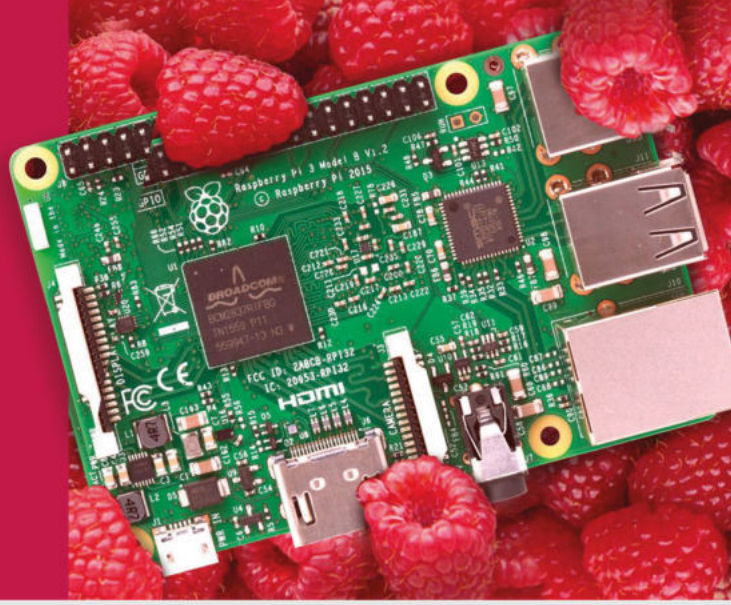
Language: **Python 3**

```
001. import time
002. import automationhat
003.
004. alarm_sounding = False
005.
006. # A short delay so you have time to get clear!
007. print("Arming in 10 seconds")
008. time.sleep(10)
009. print("Armed!")
010.
011. while True:
012.
013.     # Check the current state of the PIR sensor
014.     motion_detected = automationhat.input.three.read()
015.
016.     # If new motion detected sound the alarm
017.     if motion_detected and alarm_sounding != True:
018.         print("Motion detected, loud noise time")
019.         automationhat.relay.one.on()
020.         alarm_sounding = True
021.
022.     # If motion no longer detected stop the alarm
023.     if motion_detected != True and alarm_sounding:
024.         print("It's all gone quiet, switching off")
025.         automationhat.relay.one.off()
026.         alarm_sounding = False
027.
028.     time.sleep(0.5)
```



◀ A close-up of the PIR sensor's potentiometers. Carefully adjust with a small screwdriver as they can be fragile

American **Raspberry Pi** Shop



- Displays
- HATs
- Sensors
- Cases
- Arcade
- Swag
- Project Kits
- Cameras
- Power Options
- Add-on Boards
- Cables and Connectors
- GPIO and Prototyping

Partner and official reseller for top Pi brands:

adafruit · sparkfun ELECTRONICS · PIMORONI · HiFiBerry · and many others!

Price, service, design, and logistics support for **VOLUME PROJECTS**

USA PiShop.us

Canada BuyaPi.ca

Raspberry Pi APPROVED RESELLER

Part 01

Make beats in style with Hex-A-Pad

A stylish, capacitive touch sensor that can be used to trigger sounds, and record and replay loops, direct from Raspberry Pi

MAKER

Mike Cook

Veteran magazine author from the old days, writer of the Body Build series, plus co-author of *Raspberry Pi for Dummies*, *Raspberry Pi Projects*, and *Raspberry Pi Projects for Dummies*.

magpi.cc/mikecook

Warning!
Power tools

Please use safety goggles and gloves when drilling, sawing, and sanding

You'll Need

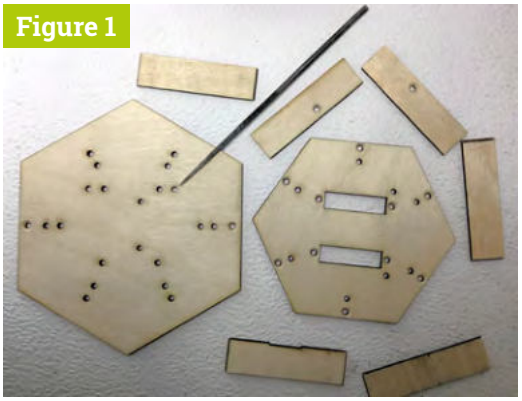
- MPR121 capacitive touch sensor **magpi.cc/mpr121**
- Strip brass 1 mm thick 50 mm by 300 mm **magpi.cc/brass1**
- 2 × sheets of birch plywood 3 mm thick A4 **magpi.cc/birch**

As you might know, we are very keen on hexagons here at the Pi Bakery, and this month we were inspired by an Adafruit project (magpi.cc/adafruitmidi). While the aesthetic looked cool, there were things we didn't like. The project required you to have access to a CNC milling machine and 3D printer, it was MIDI only, and used Bluetooth. So, we set out to design something you could make with normal workshop tools, much simpler electronics, on-board touch-sensitive controls as well as note triggers, and which would use the Raspberry Pi.

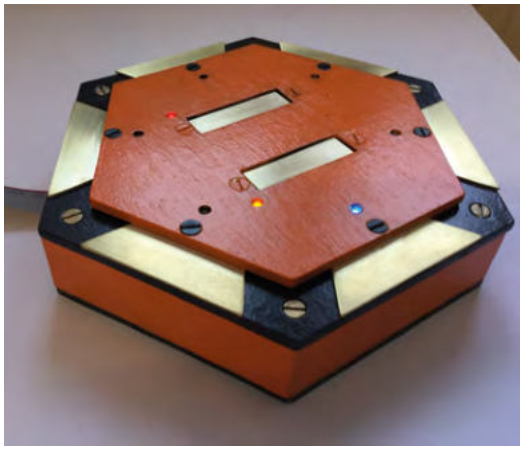
01 Workshop tools used

We used a disc sander, pillar drill, junior hacksaw, vice, and files. For cutting out the shapes of case, you can use a bench saw – we used the Proxxon KS230 circular saw – and fret-saw, or, if you have access to one, you can use a laser cutter. Either way, the files you need to use are on our GitHub page for this project. The technique for using a saw instead of a laser cutter is simple: just print out the PDF file for the part you need and glue it to the plywood. Then follow the cuts with the saw. Use a fret-saw for cutting out the central slots in the cover.

Figure 1



▲ **Figure 1** Top plate, cover, and sides. The file is pointing to one of the two tri-colour LEDs holes that might need enlarging

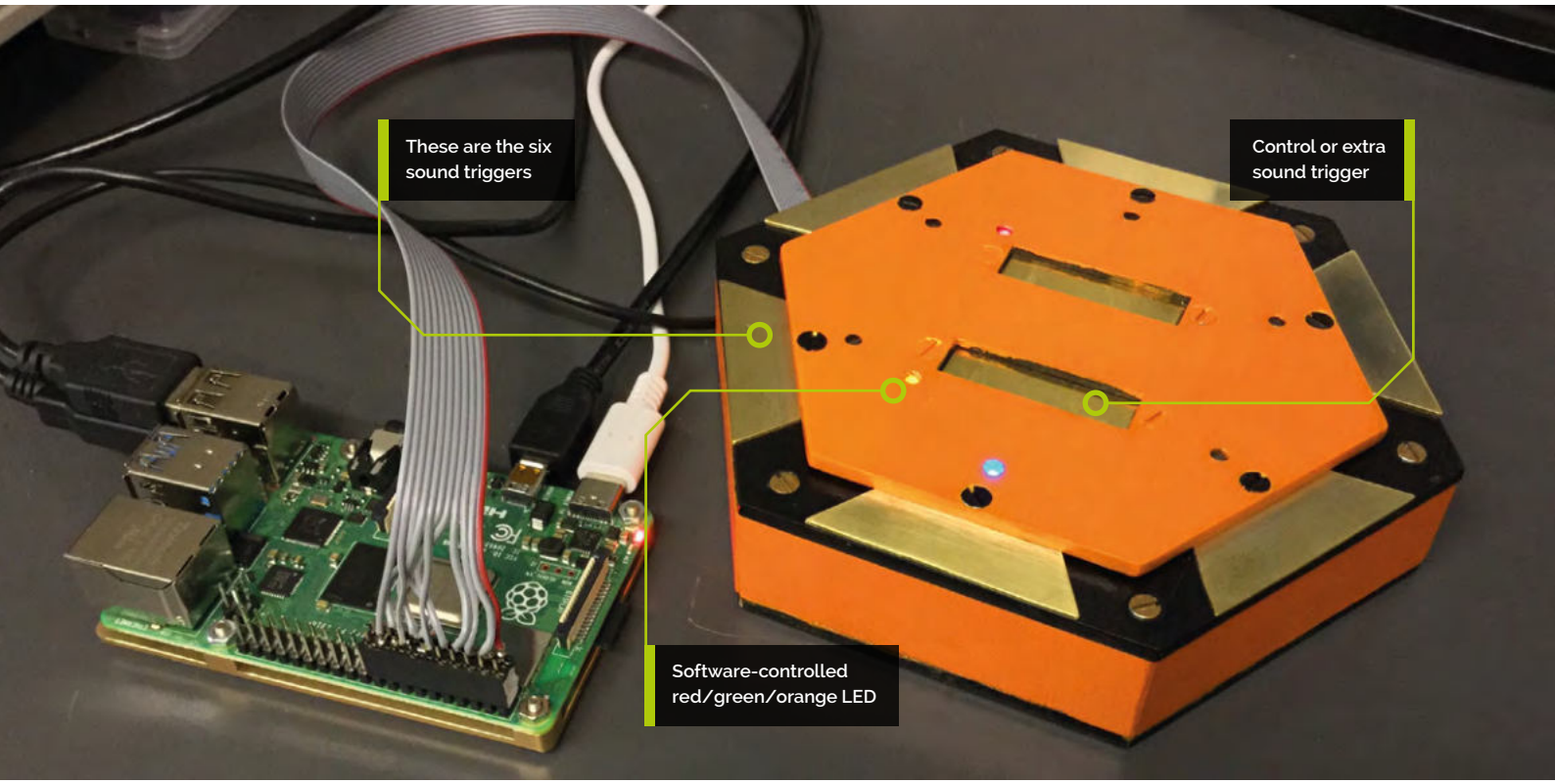


▲ The Hex-A-Pad in all its glory

02 The case parts

Wear goggles and gloves

The case consists of three hexagonal parts – base, top, and cover – along with six side panels. **Figure 1** shows all the pieces. If you are not using a laser cutter, it is important that all the holes line up exactly. So first, drill all the holes in the top plate – note that there are two 2.5 mm holes and the rest are 3 mm. Then clamp the top to the base and drill through the six holes in each of the corners of the hexagon. Countersink these holes on the underside of the base and the top of the top plate. Finally, drill the two ribbon cable clamp holes in the base.



Top Tip

The control brass strips

When cutting the control brass strips, you don't have to be too accurate with the 10 mm width because the strip can overlap the control slot.

03 Making the cover

Take the cover plate and drill two holes on opposite sides of the hexagon from the paper template. Then fasten the top plate and cover together using a nut and bolt through each hole. Next, drill through all the 3 mm holes in the top plate into the cover plate to make sure that they line up exactly. Then countersink the six holes that are in the centre of each hexagon side closest to the edge, and also the holes at the top and bottom of the slots. Finally, drill, and countersink, the two 2.5 mm holes in the top plate.

Figure 2

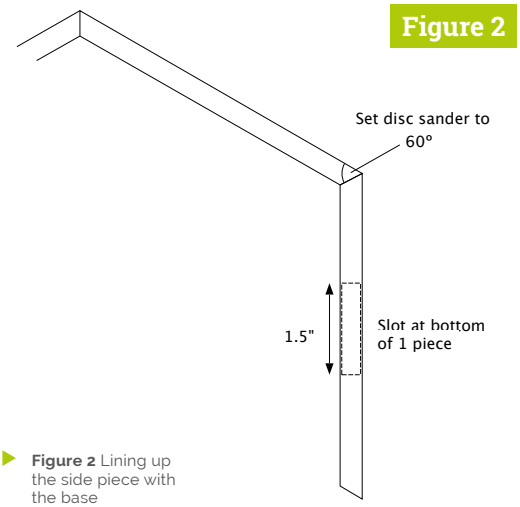


Figure 2 Lining up the side piece with the base

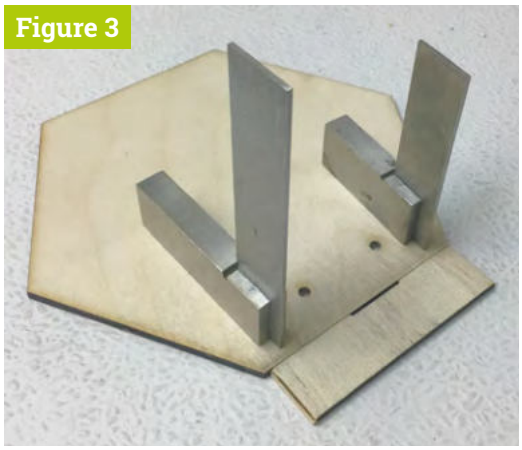
04 Making the sides

To complete the box, cut out the six sides. You will notice that one side has a very small slot in it: this is for the ribbon cable to leave the box and is only 1 mm deep – make it with a couple of strokes of a flat file. Each edge of each side piece needs to be bevelled at 60 degrees; we did this using a disc sander with the guide set to this angle using a 30/60 set square. The shape is shown in **Figure 2**; make sure the bevelled edges are done so that the side with the ribbon cable exit slot rests on the bottom of the case.

05 Fixing the sides

Each side is glued on initially using superglue. If you have used a laser cutter, it is important to sand the base edge to remove the crystallised resin from the wood, otherwise the glue will not stick. Carefully align the piece using three set squares, as shown in **Figure 3**, before applying glue to the base of the side and any adjoining side piece. Manoeuvre into place and press down (**Figure 4**). You might have to make the pieces slightly shorter if they are too big. When you have stuck all the pieces down, reinforce them with a fillet of glue around the inside edges. If you have a gap in any of the corner pieces, use wood filler and sand down for a smooth finish.

Figure 3



▲ **Figure 3** Lining up the side piece with the base

06 Making the brass inserts

The brass strip proved quite tough to work. If we had the tools, we would have annealed it to make it more workable: what is required is to heat it up until it's cherry red and then leave it to cool. Nevertheless, we managed to cut it using a new hacksaw blade and plenty of lubrication – specialist cutting oil is best, but ordinary lubricating oil will do. We cut the strip into the shapes shown in **Figure 5**; first, a slice was cut off the strip and then the 30 degree slope from each corner. Always cut on the waste side of the line and finish off with a file or disc sander.

> "A slice was cut off the strip, then the 30 degree slope from each corner"

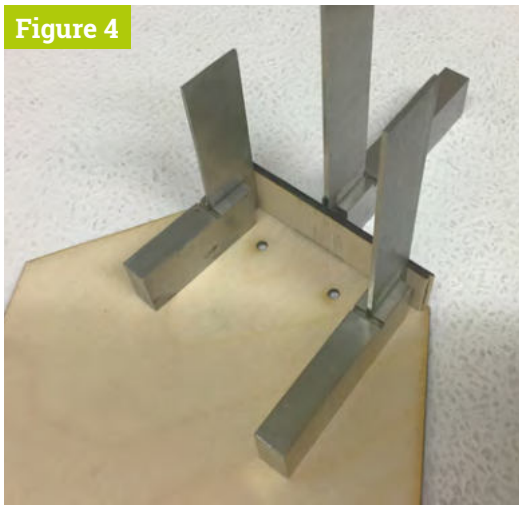
07 Painting the box

We used a combination of PlastiKote Flat Black B2 and 'Game Color' Orange Fire 72.008 to paint the box. Using a fine P600 wet and dry grit paper, we smoothed down the wood after the first of two coats. Then we applied two coats of Ronseal matt clear varnish. The tops of the six-note trigger insert screws were painted black and the four control screws were painted orange, with the six corner holes screws painted Antique Gold B34 to match the brass. Note that only a small part of the top plate is seen, therefore you need only paint the corners of the top plate.

08 Drilling the brass inserts

We marked the brass control strips through the holes in the top cover and drilled them out 3mm. Then we put the brass strips on, shown in

Figure 4



◀ **Figure 4** Gluing up the side pieces

▼ **Figure 5** The brass inserts' dimensions

Figure 5

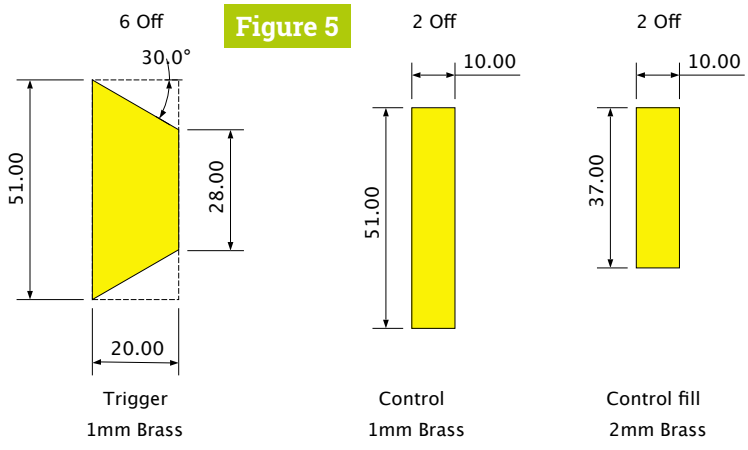


Figure 6

▲ **Figure 6** All the brass inserts

Top Tip

Painting screw tops

When painting the tops of the countersunk screws, we used a lump of Blu Tack to hold them upright. We only use these for the final assembly; before that, we used unpainted screws for the many trial assemblies.

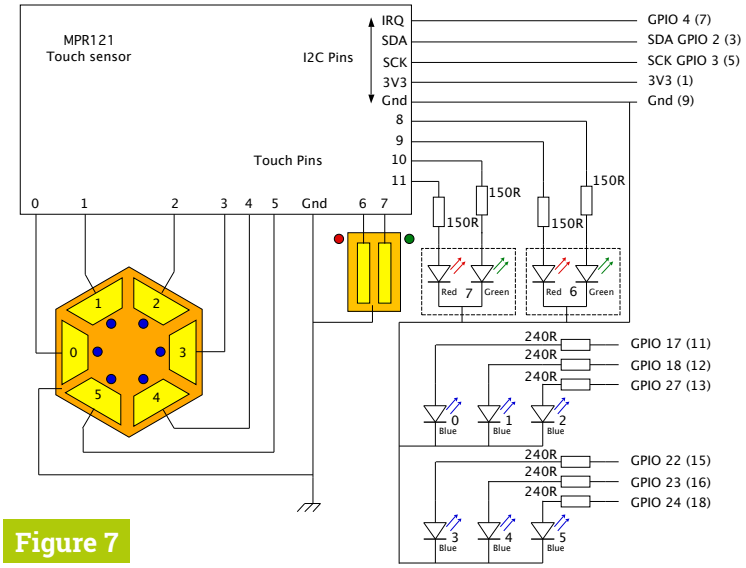
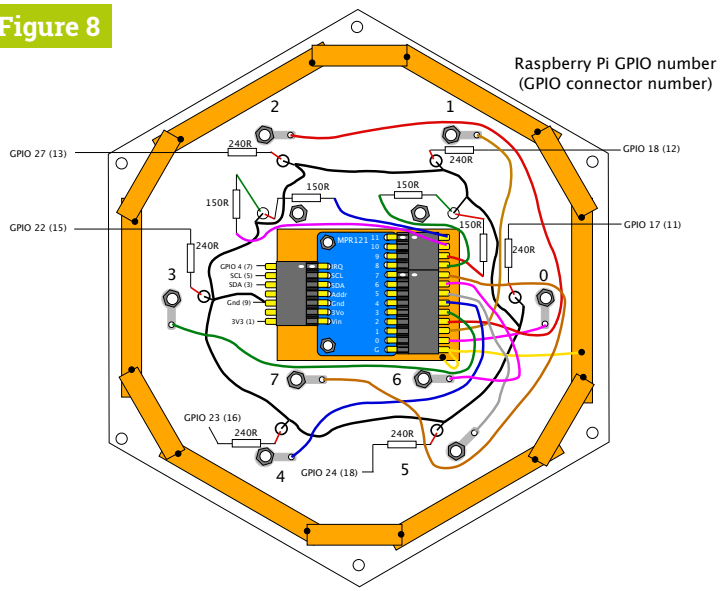


Figure 7

> The brass inserts need filing and sanding to remove sharp edges and corners

Figure 8

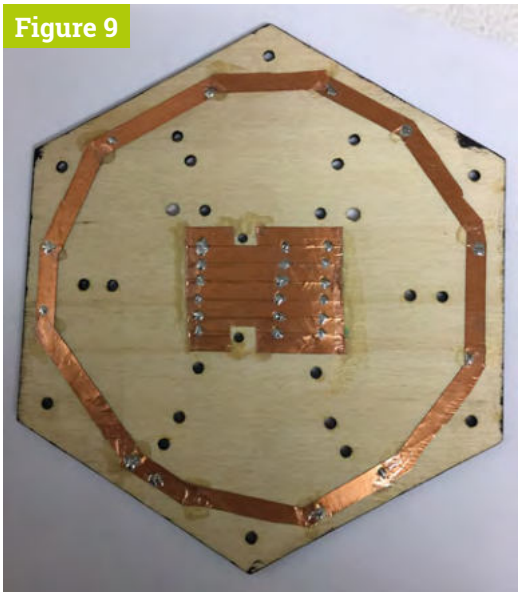


▲ **Figure 7** Schematic of the Hex-A-Pad

▲ **Figure 8** Physical layout of the Hex-A-Pad

Figure 6, and placed the top plate over them and tightened them up. Note: if the edge of the strips overlaps a portion of the LED hole, a few strokes with a rat-tail file above the hole makes sure the insert will not foul the hole. Each trapezoidal brass trigger was positioned to be at the edge of the hexagon and slotted underneath the cover, and the hole positions marked. They were then removed, drilled, and numbered with an engineer's scriber in a place that would eventually be covered up. Do not enlarge the holes too much, as we need the edge of the screw to make contact with the brass.

Figure 9



▲ **Figure 9** Grounding copper foil

09 Polish the brass

First, the brass inserts need filing and sanding to remove sharp edges and corners. Run your finger over them to check all is smooth. The brass only needs to be polished up to the mounting hole. We used a P600 wet and dry grit paper – first dry and then wet – to remove any scratches, and finished off with Brasso polish. When installing all the brass for the final time, make sure you check continuity with a meter between the brass and the screw. If it is open circuit, then put a very small amount of conducting paint in the hole and try again. Similarly, conducting paint is used to fix the rectangular control fill pieces to the control strip to bring them closer to the surface of the cover plate.

10 The schematic.

The schematic of our circuit is shown in **Figure 7**. Basically, it is the MPR121 touch sensor breakout board and a bunch of LEDs. We used 3 mm blue LEDs for the key triggers controlled by GPIO outputs, and 3 mm red/green common cathode LEDs for the control strip indicators, all in a diffused package. You might have to enlarge the two holes for these in the top plate to allow you to push them through. These were driven from four of the unused touch sensor connectors. Special registers inside this complex chip were used to make these four touch inputs into LED driving outputs.

simplesounds.py

DOWNLOAD THE FULL CODE:
magpi.cc/pibakery

Language: **Python**

```
#!/usr/bin/env python3
# Test the Hex-A-Pad touch sensors and LEDs
# with simple sounds
# By Mike Cook January 2020
import sys
import time
import board
import busio
import adafruit_mpr121
import digitalio as io
import pygame

def main():
    global last_touch
    init()
    setMPR121()
    last_touch = cap.touched()
    print ('Adafruit MPR121 Capacitive Sensor Test')
    print ('Press Ctrl-C to quit.')
    while True:
      if not capSenseNew.value:
          cur_touch = cap.touched()
          for i in range(0,8):
              readPins(i, cur_touch)
          last_touch = cur_touch
          time.sleep(0.1)

def readPins(i,c_touch) :
    global last_touch
    pin_bit = 1 << i
    if c_touch & pin_bit and not last_touch & pin_bit:
        print (i,"touched - now playing",
            soundNames[i])
        if i < 6 : LEDs[i].value = True
        if i == 7 :
            i2c.writeto(0x5A, bytes([0x78,0xC0]))
        if i == 6 :
            i2c.writeto(0x5A, bytes([0x78,0x30]))
        sounds[i].play()
    if not c_touch & pin_bit and last_touch & pin_bit:
        print (i,"released")
        if i < 6 : LEDs[i].value = False
        if i == 7 :
            i2c.writeto(0x5A,bytes([0x79,0xC0]))
        if i == 6 :
            i2c.writeto(0x5A,bytes([0x79,0x30]))

def init():
    global i2c, cap, capSenseNew, LEDs, sounds
    global pygame, soundNames
    i2c = busio.I2C(board.SCL, board.SDA)
    cap = adafruit_mpr121.MPR121(i2c)
    capSenseNew = io.DigitalInOut(board.D4)
    capSenseNew.direction = io.Direction.INPUT
    capSenseNew.pull = io.Pull.UP
    LEDs = []
    LEDpin = [board.D17, board.D18, board.D27,
              board.D22, board.D23, board.D24]
    for i in range(0,len(LEDpin)):
        led = io.DigitalInOut(LEDpin[i])
        led.direction = io.Direction.OUTPUT
        LEDs.append(led)
    cap.reset()
    # to see what you have
    print("I2C devices at",i2c.scan())
    pygame.mixer.pre_init(44100, -16, 12, 512)
    pygame.init()
    pygame.mixer.music.set_volume(1.0)
    soundNames = ["ambi_choir", "bass_voxy_hit_c",
            "drum_splash_hard",
 "drum_tom_hi_hard",
            "drum_tom_lo_hard", "drum_snare_hard",
            "bass_voxy_c", "loop_amen_full"]
    sounds = [ pygame.mixer.Sound("sounds/"+
               soundNames[i]+".wav")
               for i in range(0,len(soundNames))]

def setMPR121(): # top 4 sensor inputs to GPIOs
    # turn off cap sense
    i2c.writeto(0x5A, bytes([0x5e,0]))
    #gpio enable top 4 bits
    i2c.writeto(0x5A, bytes([0x77,0xf0]))
    # control 0 control 1 direction
    i2c.writeto(0x5A, bytes([0x73,0xf0]))
    i2c.writeto(0x5A, bytes([0x74,0xf0]))
    i2c.writeto(0x5A, bytes([0x76,0xf0]))
    # limit sensor to first 8
    i2c.writeto(0x5A, bytes([0x5e,8]))

if __name__ == '__main__':
    main()
```

Figure 10

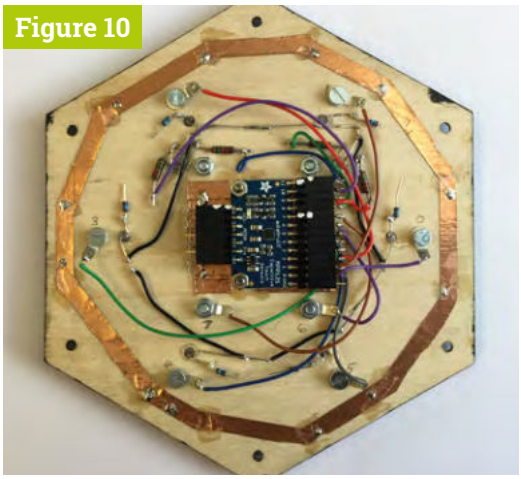
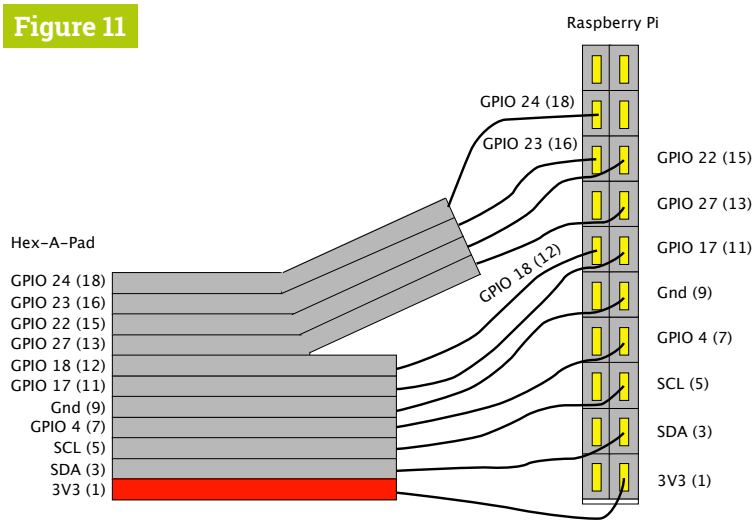


Figure 10 Photograph of wiring

Figure 11 Ribbon connections to Raspberry Pi's GPIO header

Figure 11



11 Starting the circuit

The physical layout of this circuit is shown in **Figure 8**. Start with the copper foil backing for each sensor, which provides a good ground for the sensors. **Figure 9** shows a photograph of the foil, with each strip connected by a small solder blob. The foil was also kept in place using some blobs of transparent Gorilla glue. Note the cut-out for the touch sensor board screws; this was fitted on M2.5 screws and 5 mm M3 nylon spacers. Note that the board was fitted with right-angle pins to keep a low profile. The LEDs are pushed through the holes into the cover, leaving them flush with the top surface.

12 Building the circuit

The resistors were mounted flat on the board to the LEDs and, when all were wired up, were held down in place by spots of hot-melt glue in the LED holes. Then the touch sensor wires were connected as shown in **Figure 10**. Finally, the connections to the ribbon cable were made – shown in **Figure 11**, along with where they connect to a 20-pin header which fits onto the first pins of Raspberry Pi's GPIO. Note: pass the ribbon connector through the slot in the base before soldering up the 20-pin header socket. A small wooden bar with foam glued on the back of it stops the ribbon cable being pulled out.

Top Tip

Installing CircuitPython

In a command-line Terminal, type the following:

```
sudo apt update
sudo pip3 install --upgrade setuptools
sudo pip3 install adafruit-blinka
sudo pip3 install adafruit-circuitpython-mpr121
```

> A small wooden bar with foam glued on the back of it stops the ribbon cable being pulled out

13 Testing software

The **simplesounds.py** listing shows some simple code that allows you to test all the LEDs and trigger some simple sounds. There is a sound for each of the six note triggers, as well as the two control bars. The sounds were taken from the Sonic Pi sample files, found at **/usr/share/sonic-pi/samples**, and converted into WAV files using Audacity. We will go more into the details of what is needed next month, but this software requires the installation of Adafruit's CircuitPython framework along with the MPR121 driver. We had no trouble installing this under the Raspbian Buster operating system, but found when we used earlier versions of the OS it did not install correctly, throwing up errors.

Next month, we'll look at software to drive the touch sensor chip in a more reliable way and also a fully fledged MIDI / sound sample keypad with effects control, sequence recording, and playback.

3 ISSUES FOR £5

Subscribe by phone: **01293 312193**

Subscribe online: **magpi.cc/subscribe**

Email: **magpi@subscriptionhelpline.co.uk**

Part 10

Glade layout editor with C and GTK

Use this layout editor to more easily create window layouts for applications

MAKER

Simon Long

Simon Long is a software engineer working for Raspberry Pi, responsible for the Raspberry Pi Desktop on both Raspbian and Debian.

rpf.io

As can be seen in the previous tutorials in this series, while creating widgets with GTK is a lot easier than doing it all from scratch by writing pixels to screen buffers, you can still end up with quite a lot of code for even a fairly simple window layout, and you do have to think quite carefully about the code you are writing to make sure the window looks the way you want it to. It's also less than ideal that you can't actually see what the window will look like until you run the application.

Fortunately, there is a way to work on the layout of a window without having to write all its code, and that is to use a *layout editor*. GTK allows the layout of widgets on a window to be defined in an XML file, which is then loaded and drawn on the screen when the application runs – you still need to link up the behaviour of any widgets to code, but the appearance of the widgets can be determined in advance. A layout editor is a useful tool to create such an XML file.

The most widely used layout editor for GTK is a tool called Glade. There are two versions of Glade, one for GTK 2 applications and one for GTK 3 applications; they aren't interchangeable, so you need to use the correct one. The version to create GTK 3 applications is called 'glade'. And, just to be perverse, the version you need to create GTK 2 applications is called 'glade-3'. (Yes, that is counter-intuitive and quite annoying!)

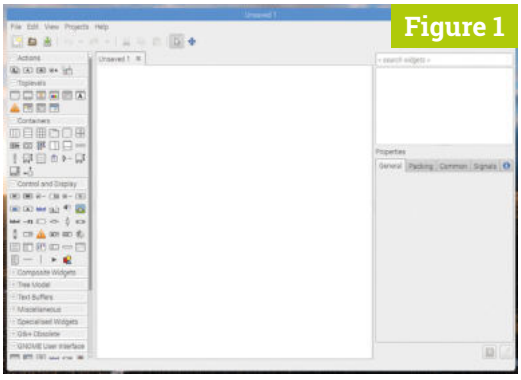
The first thing we need to do is to install glade-3. At a Terminal, type `sudo apt-get install glade-3`, and answer yes to any prompts. You should then have an entry labelled 'Glade' in the Programming menu on your desktop. If you launch it, you'll see a screen like the one shown in **Figure 1**.

This is the basic design screen in Glade. On the left side is a palette of all the widgets available. In the centre is the area where you actually design your window, and on the right is an area which allows you to set the properties of individual widgets.

Using Glade to create a layout file

We'll create a simple window for our application, so the first thing we need to do is to add a top-level window. Under the palette section labelled 'Toplevels', the top-left entry is a GtkWindow – if you hover over it with the mouse, a tooltip will pop up with the label 'Window' to confirm this. So just click on this icon to add a blank window to the design area in the centre of the screen (**Figure 2**).

Figure 1



▲ **Figure 1** The Glade layout editor

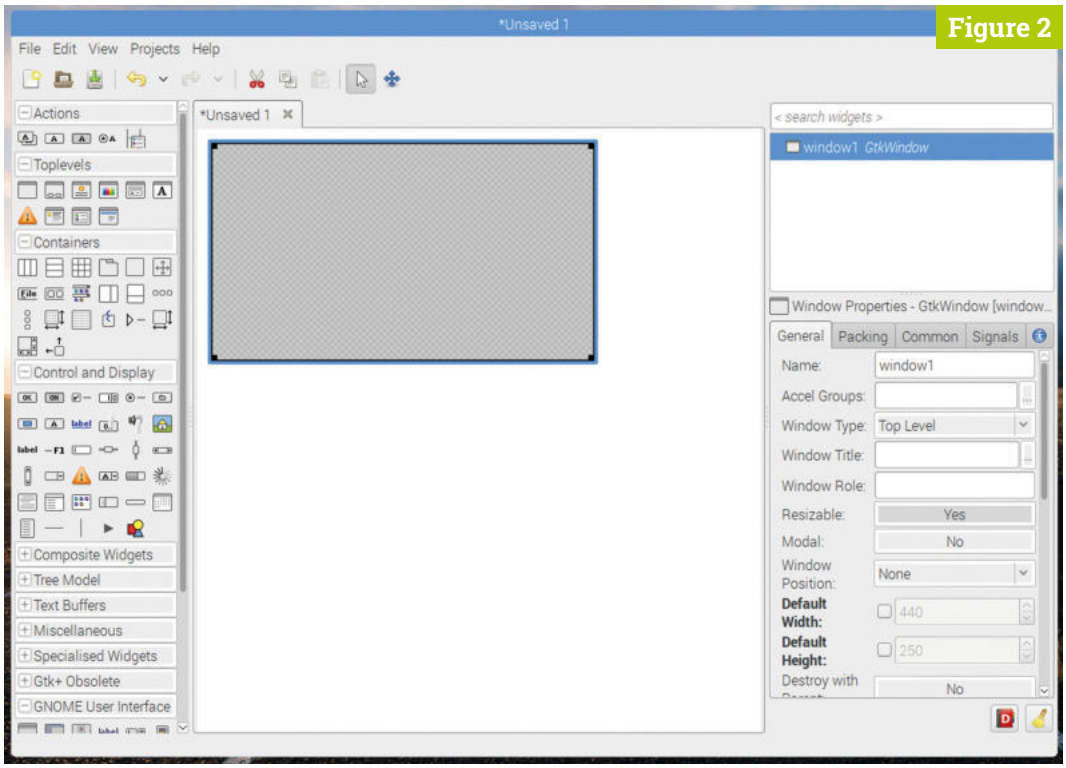


Figure 2

Figure 2 An empty window

Let's add a VBox to put some widgets in. The widget palette has a section labelled 'Containers', in which the first two entries are an HBox and a VBox – click the VBox, and then move the cursor into the blank window and click within it. A dialog will pop up asking how many items you want in the VBox – set it to 2, and click 'Create' (**Figure 3**, overleaf).

You've now got a blank window with a line across the middle, showing the VBox. We're going to add a label to the top, and a button to the bottom. Find these in the 'Control and Display' section of the palette – first click on each in the palette, and then click in the relevant section of the VBox in the design area to add them (**Figure 4**).

If you look at the top of the right-hand column, you'll see the hierarchy of widgets being displayed – a GtkWindow, containing a GtkVBox, containing a GtkLabel and a GtkButton. If you click on one of the widgets in the hierarchy, the options displayed in the tabbed area at the bottom of the right-hand column will be those for that widget.

Click on the GtkLabel, and scroll through the options on the 'General' tab; you'll find a section labelled 'Edit label appearance', including the entry 'Label' – use this to set the text in the label to whatever you want (**Figure 5**).

Click on the GtkButton, and find the Label section for it – it's under 'Label with optional image' – change the label for the button to 'Quit'.

It's worth investigating the options on all the tabs at the bottom of the right-hand column – there are too many to go into here, but of particular interest are those on the 'Packing' tab; these allow you to set the expand, fill, and padding options which were available when adding widgets to boxes from code – if you play with them, you can get a good feeling for how much control you have over layout of the window, particularly when combining multiple boxes inside each other, as Glade allows you to do.

> If two widgets have the same name, the layout file is invalid and will not load correctly

The other thing that it is important to note about the right-hand column is that Glade assigns every widget a name – it can be seen in the hierarchy list at the top, and as the top entry on the 'General' tab. Glade assigns names as the name of the widget type followed by a number, but you can change these if you want. But there is one *vital* rule – every widget in a layout file must have a unique name. If two widgets have the same name, the layout file is invalid and will not load correctly.

Your button and label widgets should have the names 'button1' and 'label1', and your window should be 'window1' – remember these, as they are important when linking the code to the layout file. For now, save your layout as **mylayout.glade** by choosing the 'Save As' option in the 'File' menu.

Figure 3

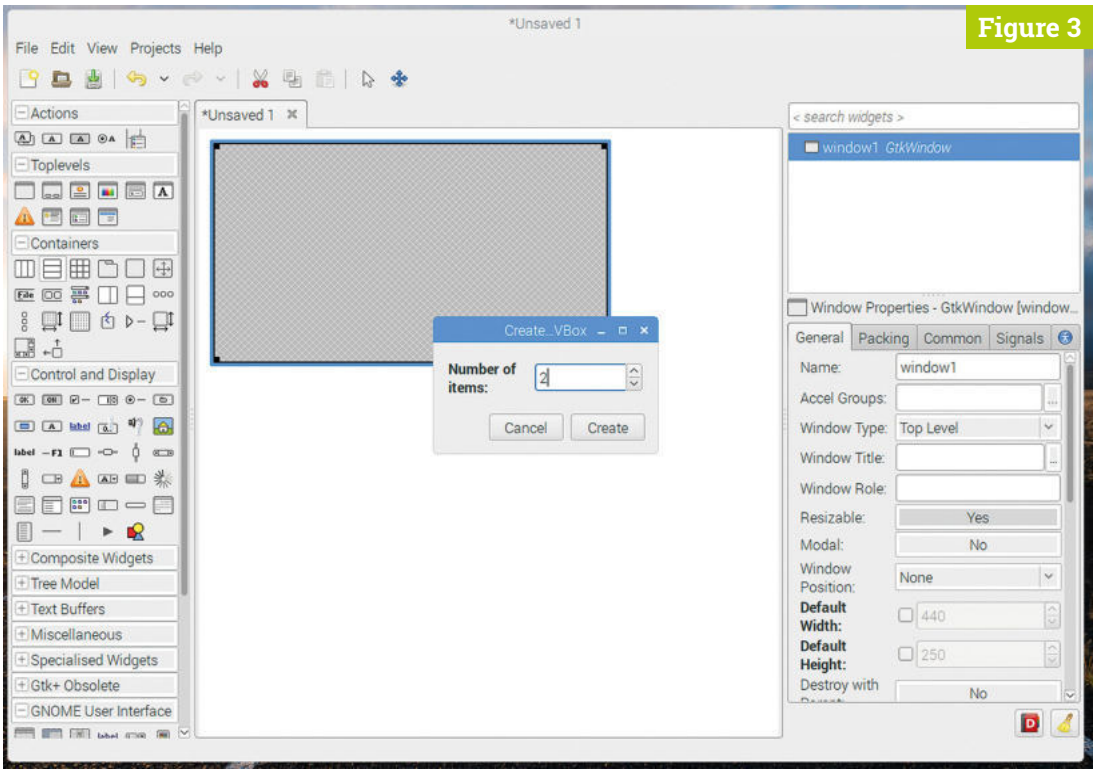


Figure 3 Adding a vertical box

(You can select one of two different formats on the save window – use the default GtkBuilder option.) Make sure you save the file in the same directory where you are storing your GTK source code.

Using a layout file in a GTK application

Now we need to load the layout file into some GTK code. Try the following:

```
#include <gtk/gtk.h>

void main (int argc, char *argv[])
{
  gtk_init (&argc, &argv);

  GtkBuilder *builder = gtk_builder_new ();
  gtk_builder_add_from_file (builder,
      "mylayout.glade", NULL);

  GtkWidget *win = (
      GtkWidget *) gtk_builder_get_object (
      builder, "window1");
  gtk_widget_show_all (win);
  gtk_main ();
}
```

The `gtk_init` and `gtk_main` calls are the same as we have seen before, but the code between them uses a GtkBuilder, a code object which reads in and processes a layout file.

We create a GtkBuilder object:

```
GtkBuilder *builder = gtk_builder_new ();
```

We then read the layout file into the builder.

```
gtk_builder_add_from_file (builder,
    "mylayout.glade", NULL);
```

The widgets we created in the layout file can now all be accessed by calls to `gtk_builder_get_object`, and used as before. So we get the object which was named `window1` in the layout file.

> To make the widgets actually do things, we need to connect signals

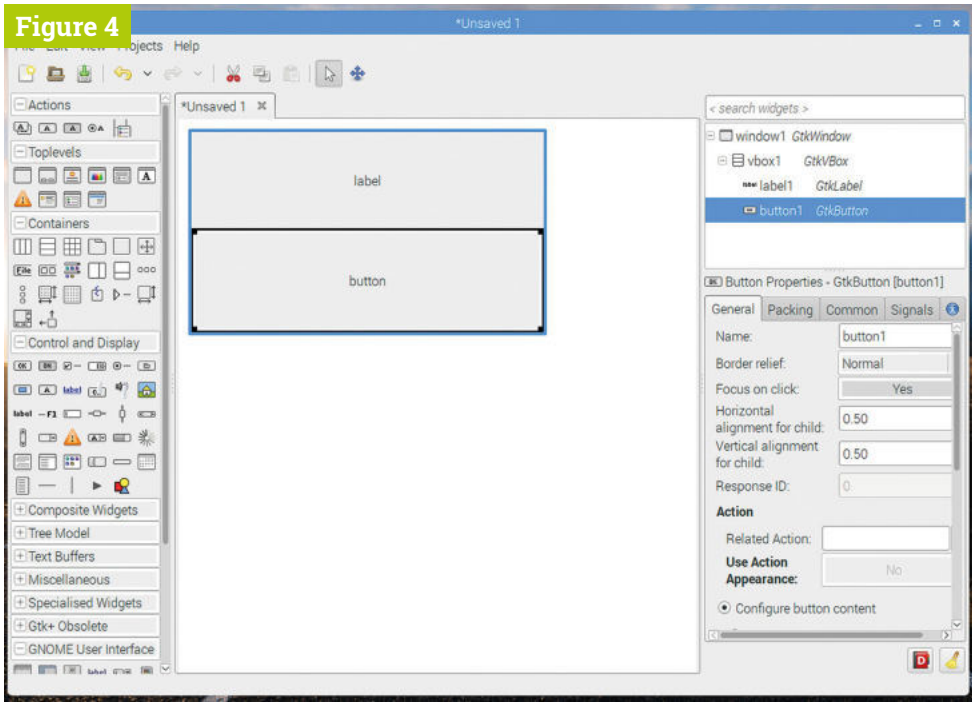
```
GtkWidget *win = (
    GtkWidget *) gtk_builder_get_object (
    builder, "window1");
```

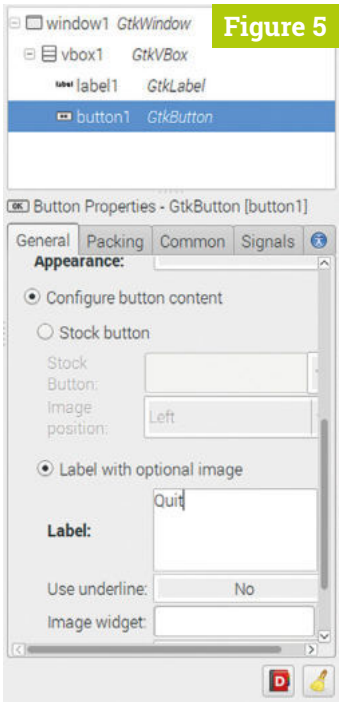
And we show it as before:

```
gtk_widget_show_all (win);
```

If you build and run this code, our simple window should be displayed on the screen (**Figure 6**).

To make the widgets actually do things, we need to connect signals as before, so let's modify our code accordingly.





◀ **Figure 4** With a label and a button added

▲ **Figure 5** The widget properties editor

```
void end_program (GtkWidget *wid,
    gpointer ptr)
{
  gtk_main_quit ();
}

void main (int argc, char *argv[])
{
  gtk_init (&argc, &argv);

  GtkBuilder *builder = gtk_builder_new ();
  gtk_builder_add_from_file (builder,
      "mylayout.glade", NULL);

  GtkWidget *win = (
      GtkWidget *) gtk_builder_get_object (
      builder, "window1");
  GtkWidget *btn = (
      GtkWidget *) gtk_builder_get_object (
      builder, "button1");
  g_signal_connect (btn, "clicked",
      G_CALLBACK (end_program), NULL);

  gtk_widget_show_all (win);
  gtk_main ();
}
```

This time, we also get the button widget from the builder and connect the `end_program` callback to it with `g_signal_connect`, as before. Build and run, and this time the button should close the program.

Using layout files can make the amount of code you need to write a lot shorter, and the use of Glade makes it easier to create complex interfaces with more customisation of your widgets. For small applications with a simple window, it's probably more straightforward just to do everything in C, but for anything more complex, using a layout file can make the code a lot more readable.

Next steps

One thing which should be obvious to anyone who has worked through this book and reached this point is that GTK is a fairly huge subject! There are dozens of widgets, and hundreds of different ways they can be configured and used – any comprehensive guide to using them all would run into several hundred pages.

The best reference to GTK 2 – and the best place to find further information if you want to do more with it – is the official developer documentation, which can be found online at **magpi.cc/gtk2manual**. This gives far more detail on every aspect of GTK – the reference section listing all the functions associated with each widget is particularly useful.

Hopefully this series of tutorials has given you an idea of where to get started with GTK programming, and enough of a foothold to be able to experiment for yourself.

Happy GUI designing!

▼ **Figure 6** The window created in Glade when run

NEW YEAR NEW PROJECTS!

Get inspired to make something in 2020

A new year means new plans. Sometimes it's not easy to come up with a big cool thing to work towards though, especially when it comes to making.

We're here to help. Whether you're making your first project or wanting to try something new, we think we have you covered with a selection of projects to try, or just get inspired by.

If you need help figuring out what tools and accessories you'll need for all this, make sure to check out issue 89 (**magpi.cc/89**) for some big features that will help you out. Otherwise, let's get started.

FUN PYTHON PROJECTS

Improve your coding skills

GET STARTED WITH THE TWITTER API

Automate your tweets with Python

Working with Python is fairly easy as programming languages go. You can do a lot more with it than just have it say Hello World, though, such as using it in conjunction with social media network Twitter!

In this tutorial, you'll learn how to work with APIs – a way to interface with other programs and services – using Python and Twython, with the result being you can send tweets via Python code. This includes tweets with photos attached, and allows you to view a feed of tweets as well. Great for robots or online bots.

magpi.cc/twitterapi

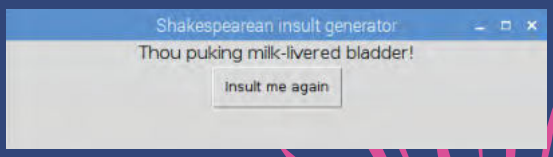
SHAKESPEAREAN INSULT GENERATOR

Make cursing fun

Python can be used for practical, interesting projects; however, it's always important to have a bit of fun, like with this Shakespearean insult generator.

It will teach you about some basic programming concepts, such as reading and writing data in CSV format using Python, manipulating lists, choosing random items from a list, and how to create a basic graphical user interface (GUI). These GUI skills can easily be translated to a lot of different Python projects and make them much easier to use.

magpi.cc/shakespeareaninsults

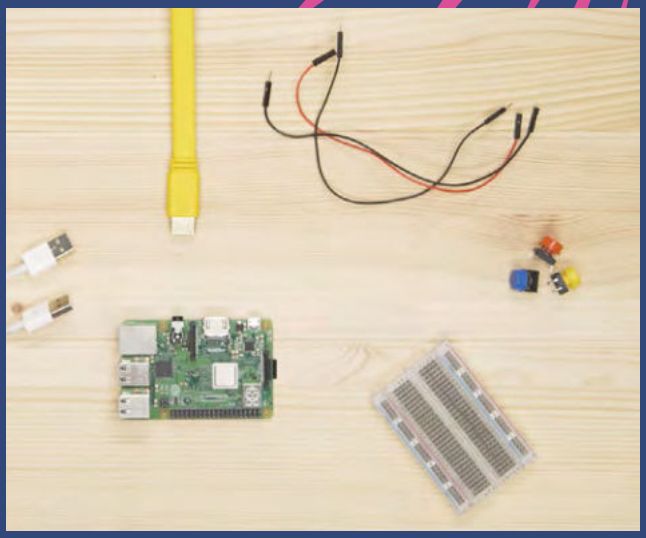


QUICK PHYSICAL PROJECTS

Bring your code into the real world

GPIO MUSIC BOX

Learn how buttons and other input components work with Raspberry Pi



Coding is fun, but have you ever used Python code to interact with real-life objects? Using a simple push-button hooked up to the GPIO pins of a Raspberry Pi, you can control the way code runs. In this project, you'll learn how to program with a push-button so that specific button presses will play different sounds.

This project makes use of both GPIO Zero – a library specifically created to make accessing components on GPIO easy – and Pygame, which can be used for programming video games in Python.

magpi.cc/gpiomusicbox

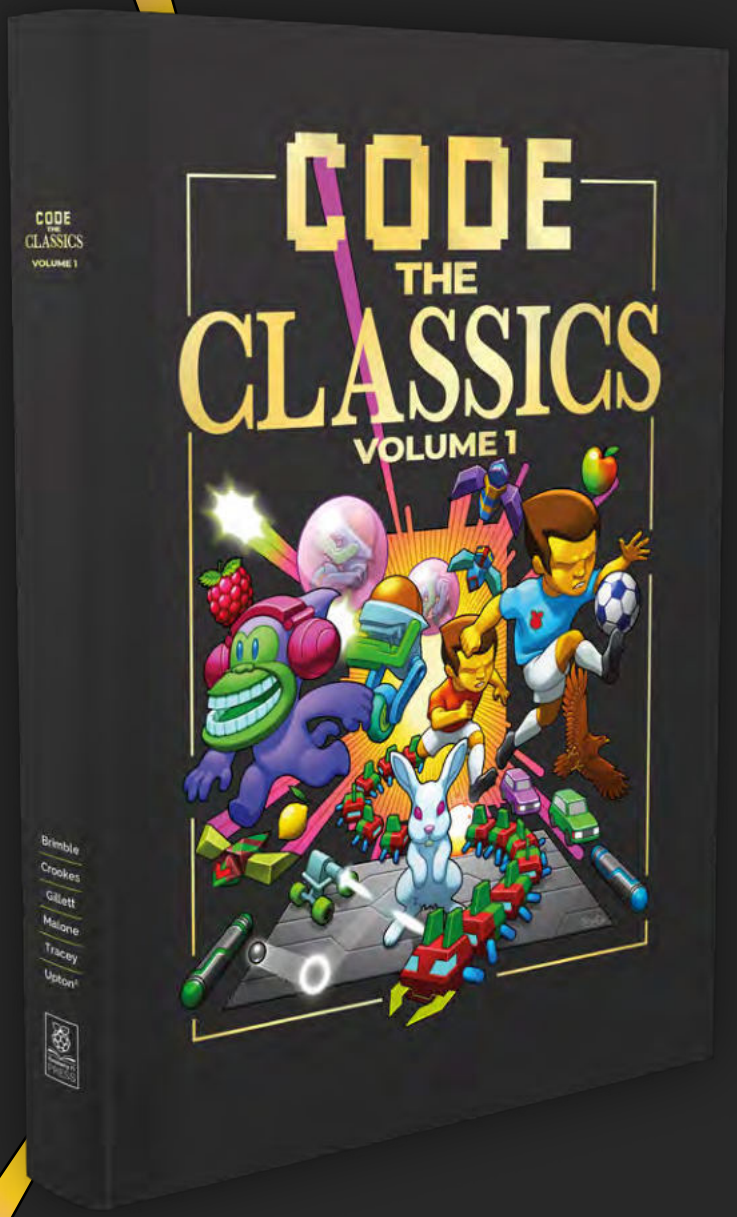
WHOOPI CUSHION

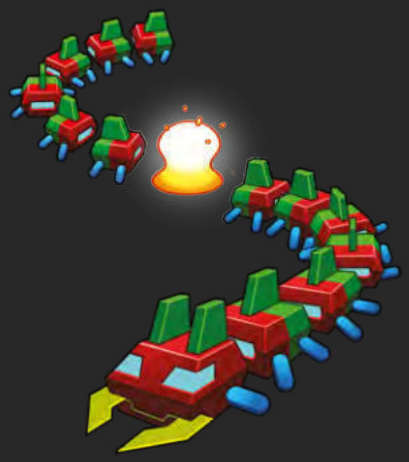
Build a very custom button to prank your friends

We could very easily lie here, without any reservations, that the reason we like this tutorial is because of the ingenious, DIY button that makes up a major part of it. However, it's mainly just an excuse to make a digital whoopee cushion, which is hilarious.

To be fair, we do also like the very DIY button made out of paper plates and tin foil and clips, and it's a great way to learn how you can bodge together your own buttons and sensors and such, while also pranking your parents, siblings, friends, roommates, or even your boss.

magpi.cc/whoopi





- ***Get game design tips and tricks from the masters***
- *Explore the code listings and find out how they work*
- ***Download and play game examples by Eben Upton***
- *Learn how to code your own games with Pygame Zero*

This stunning 224-page hardback book not only tells the stories of some of the seminal video games of the 1970s and 1980s, but shows you how to create your own games inspired by them using Python and Pygame Zero, following examples programmed by Raspberry Pi founder Eben Upton.

Available now: **magpi.cc/store**

SIMPLE CAMERA PROJECTS

Do fun stuff with the Raspberry Pi Camera Module

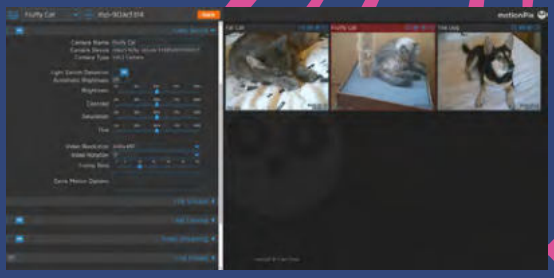
MINECRAFT PHOTOBOOTH

Take real pictures using Minecraft Pi

Minecraft on Raspberry Pi is quite different to other versions of Minecraft in that it is designed to be hacked and modded through Python code. We particularly like this Minecraft project as it lets you use Python to bridge the gap between a virtual Minecraft world and the real world outside it.

With a Raspberry Pi Camera Module and a little code, you can take pictures of the real world using a virtual button in Minecraft. There are even ways to display your photo in Minecraft if you really want to challenge yourself.

magpi.cc/photobooth



THE PARENT DETECTOR

A motion-sensing camera that might help protect your belongings

A parent detector for kids and a CCTV for the grown-ups, this simple camera project allows you to control a camera using a PIR motion-sensing component.

This version is very simple, but you can also add many features in the long run – like a live web interface, backup camera-based motion sensing, and even motion-tracking motor control if you really fancy going all-out on the project.

While this project uses Python and Raspbian, MotionEyeOS is an operating system for Raspberry Pi specifically built for CCTV stuff if you find the way this project works a little limiting.

magpi.cc/parentdetector

ROBOTS FOR BEGINNERS

Build your own automaton

CAMJAM EDUKIT 3 WITH CHASSIS

An upgrade to the regular CamJam EduKit robot

We talk about the CamJam EduKit 3 a lot – because it's very good and very cheap. One thing that we think really elevates the build is this 3D-printed chassis that turns the kit into a more recognisable programmable robot.

The CamJam kit itself is also just a good way to learn about simple robotics with Raspberry Pi, and was created by the people that run Pi Wars, so it definitely has a very good pedigree. While it may not be what you use to enter Pi Wars, it's definitely a step towards it.

magpi.cc/edurobotchassis

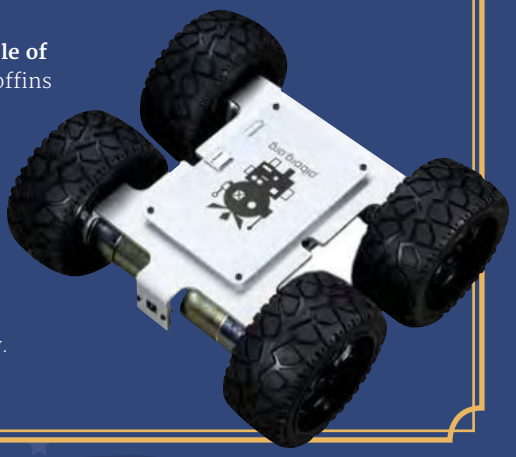
MONSTERBORG

Build, program, race, and extend with the ultimate Raspberry Pi robot kit

The CamJam kit is great for beginners, but the MonsterBorg kit is fantastic for people with a couple of robots under their belt. This robot is made by the boffins at PiBorg and used in the Formula Pi autonomous robot racing championship. This means it's very quick, very sturdy, and very hackable. It can even flip over and drive upside down.

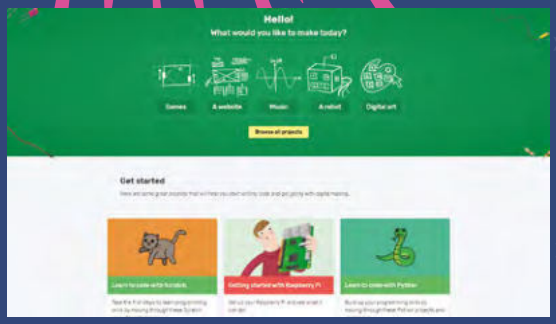
With some extra modification, you can add more sensors and kit to the robot; with the great programming libraries available, it's easy to incorporate them into your code, whether you're remote-controlling it or having it run autonomously.

magpi.cc/monsterborg



GET SOME INSPIRATION!

Where to find other projects and tutorials



RASPBERRY PI PROJECTS

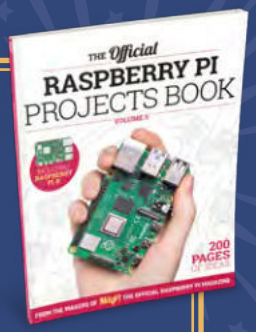
The Raspberry Pi Projects website is a massive resource for all things Raspberry Pi, from setting up your first Raspberry Pi to making robots and real-life photo-booths. You can filter by hardware, components, programming languages and more, and it's a great way to learn just how much your Raspberry Pi can do.

magpi.cc/projectsonline

INSTRUCTABLES

A veteran maker website that is chock-full of incredible, step-by-step projects that you can try to follow along to. You can find something for just about any type of hardware, and it's easy to drill down to specific types of projects, or even just see what popular Raspberry Pi projects are on there.

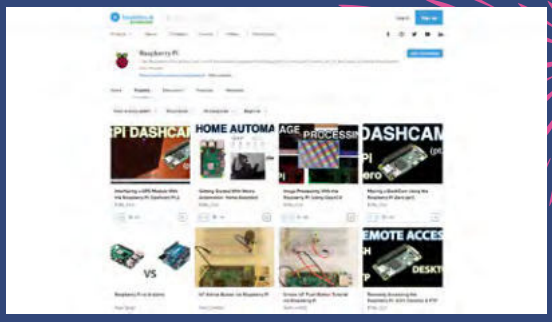
instructables.com



THE OFFICIAL RASPBERRY PI PROJECTS BOOK VOL. 5

The fifth volume of our *Official Raspberry Pi Projects Book* contains 200 pages of amazing projects and tutorials that should help give you a huge boost of inspiration. There's also a dedicated Raspberry Pi 4 beginner's section if you're brand new to Raspberry Pi.

magpi.cc/projects-book5



HACKSTER

Hackster is a newer site, but it covers a similar selection of projects to Instructables, albeit with a stronger bend towards learning computing skills. You can still filter project types, but it's easier to differentiate them by difficulty to make sure you don't quite get in over your head.

hackster.io

Wireframe

Join us as we lift the lid on video games

Visit **wfmag.cc** to learn more

GROUP TEST

Raspberry Pi thermal cases

These cases keep Raspberry Pi 4 feeling, and looking, cool. By **Gareth Halfacree**

Raspberry Pi 4 – like all the other members of the ever-growing Raspberry Pi family – is entirely usable as is, and plenty of people appreciate the aesthetic of a bare board on a desk.

For those who don't, there are a wealth of cases – both first- and third-party – available. You'll find one, the Raspberry Pi 4 Stand, mounted on the cover of this magazine, while the others in this group test can be found at all major retailers.

Each case here has been tested for aesthetics, complexity of assembly, and its performance in keeping Raspberry Pi 4 running cool.

How we tested

Each case was given a heavy synthetic workload to represent a worst-case scenario. This workload, which stresses both the central and graphics processors, runs for ten minutes followed by a five-minute cooldown period. Full details of the workload can be found in *The MagPi* issue 88, at **magpi.cc/88**.

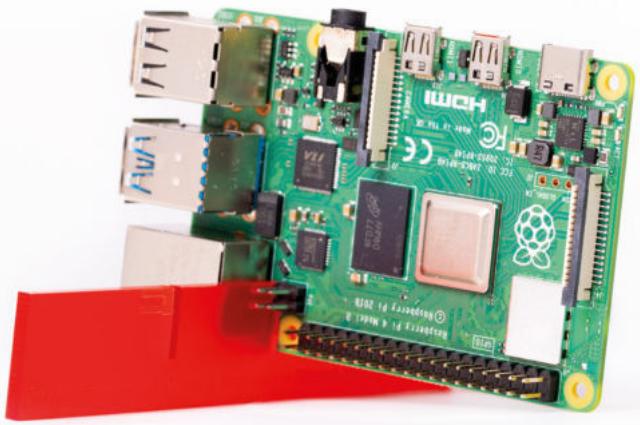
1. Raspberry Pi 4 Stand
2. Flirc Raspberry Pi 4 Case
3. Argon One for Raspberry Pi 4
4. Sensoreq CooliPi Heatsink and Case
5. Pimoroni Aluminium Heatsink Case for Raspberry Pi 4
6. The Pi Hut Raspberry Pi 4 Case 2.0

Raspberry Pi 4 **Stand**

▶ Free with issue 90 of The MagPi magazine

Made from a single piece of acrylic, the Raspberry Pi 4 Stand is as pure as it gets

The Raspberry Pi 4 Stand is about as simple as a case could possibly be. Laser-cut from a single piece of acrylic, there's no complex assembly required: simply slot the stand between the Power over Ethernet (PoE) header and Ethernet port of Raspberry Pi 4 and pop it on your desk.



The stand is designed to improve cooling by aligning Raspberry Pi 4 vertically, rather than flat on a desk. Previous thermal testing in issue 88 showed this is surprisingly effective, and the Raspberry Pi 4 Stand solves the stability issue which comes from balancing the board on its edge.

There's a bonus trick up the Raspberry Pi 4 Stand's sleeve, too: it holds up to three Raspberry Pi 4 boards side-by-side, making a very cost-effective computing cluster. Whether you install one, two, or three boards, the Raspberry Pi 4 Stand is surprisingly stable and not unattractive – and it retains access to all ports and headers.

> The Raspberry Pi 4 Stand holds up to three boards side-by-side, making a very cost-effective cluster

SPECS

DIMENSIONS:
120×20×2.8 mm

MATERIAL:
Acrylic

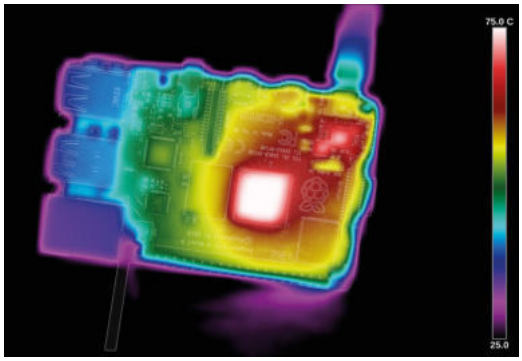
WEIGHT (INC. ONE RASPBERRY PI 4):
54 g

NUMBER OF BOARDS SUPPORTED:
Up to 3

COOLING METHOD:
Vertical alignment

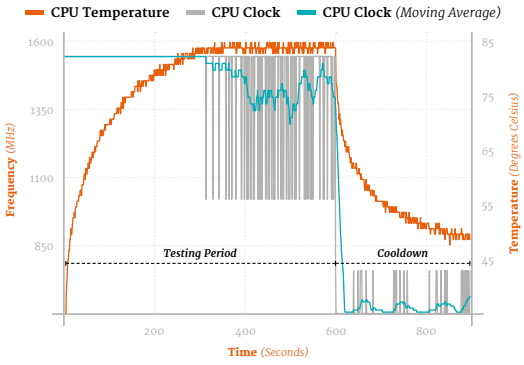
Thermal imaging

The Raspberry Pi 4 Stand improves the bare performance, but Raspberry Pi 4 still gets hot under sustained synthetic load.



Thermal load

Without additional cooling, the Raspberry Pi 4 Stand can't prevent Raspberry Pi 4 from hitting its throttle point during testing.



Verdict

The Raspberry Pi 4 Stand is smart, free, and the only case on test to support more than a single board. Its cooling performance, though, is the weakest.

Note!
We don't score our own products. [We think our Raspberry Pi 4 Stand is perfect – Ed.]

Flirc Raspberry Pi 4 Case

▶ Flirc ▶ **magpi.cc/flirc** ▶ £16 / $16

Designed to blend in with home theatre products, the Flirc case is undeniably attractive

SPECS

DIMENSIONS:
93.7×66×26.5 mm

MATERIAL:
Aluminium

WEIGHT (INC. ONE RASPBERRY PI 4):
134 g

NUMBER OF BOARDS SUPPORTED:
1

COOLING METHOD:
Passive heatsink (SoC only)

EXTRAS:
Thermal transfer material pad

Created as a means of drawing attention away from Raspberry Pi 4 when used as part of a home theatre installation, the Flirc combines a matte-finish silver aluminium housing with soft-touch black plastic to the top and underside. It's an understated design, but one which does compromise efficacy: the plastic lid covers much of the surface area of the aluminium case, reducing its ability to bleed off heat.

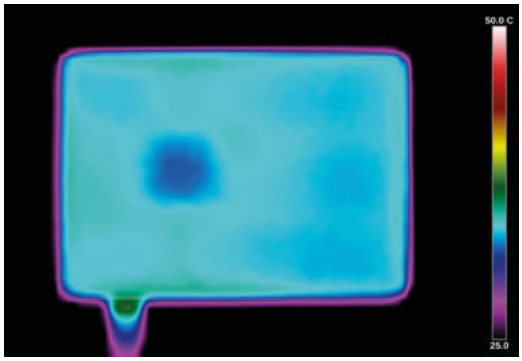
The case itself makes contact with Raspberry Pi 4's system-on-chip (SoC) via a single hollow pillar and a bundled thermal interface material pad. Installation is simple, requiring only two protective sheets to be removed from the pad, and four screws to hold the case together.

> Offers no ready access to the GPIO, CSI, or DSI headers

For those not interested in attractive home theatre setups, though, the Flirc comes with a major drawback: it offers no ready access to the GPIO, CSI, or DSI headers, though all external ports are easily reached.

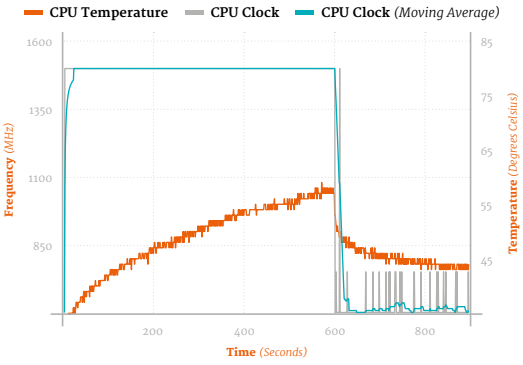
Thermal imaging

The plastic lid prevents the Flirc from cooling entirely efficiently, while the hollow pillar can be seen as a cooler spot to the centre-left.



Thermal load

Even with the lid in place, the Flirc case easily cools Raspberry Pi 4 during the synthetic workload run.



Verdict

Unless you need the GPIO, CSI, or DSI headers, the Flirc's few design flaws are unlikely to matter: the case keeps Raspberry Pi 4 well clear of its thermal throttle point.

8/10

Argon One for **Raspberry Pi 4**

▶ Argon 40 ▶ **magpi.cc/argon1** ▶ £19 / $25

Impressively feature-packed, the Argon One offers a lot for your money – including temperature-controlled active cooling

The Argon One case packs a whole lot of functionality into a surprisingly small footprint. A daughterboard connects to Raspberry Pi 4's AV and HDMI ports to re-route these to the rear of the case, alongside Ethernet and USB, while a second board pulls the GPIO header out to a colour-coded and silk-screen labelled header hidden under a magnetic cover on the top.

> Includes a smart power button to safely turn Raspberry Pi 4 on and off

The same board powers a fan, which is active when the temperature exceeds a user-configurable limit, and includes a smart power button which can safely turn Raspberry Pi 4 on and off with a press. There's even space to route out CSI and DSI cables for a camera or display.

Cooling performance is impressive. The Argon One prevented Raspberry Pi 4 from throttling without even needing to activate the fan – aided by the entire aluminium surface acting as a heatsink for the SoC and RAM chips.

SPECS

DIMENSIONS:
105×95.6×35 mm

MATERIAL:
Aluminium

WEIGHT (INC. ONE RASPBERRY PI 4):
230 g

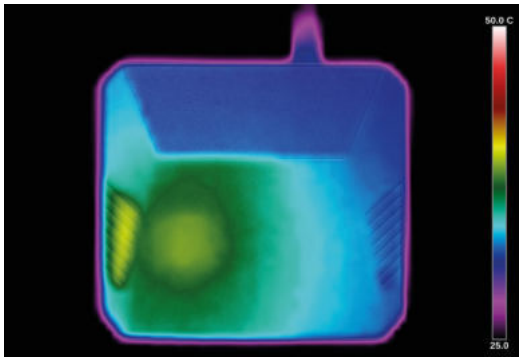
NUMBER OF BOARDS SUPPORTED:
1

COOLING METHOD:
Passive heatsink (SoC, RAM), PWM fan

EXTRAS:
Thermal transfer material pads, AV daughterboard, fan, labelled GPIO header with magnetic cover, smart power board

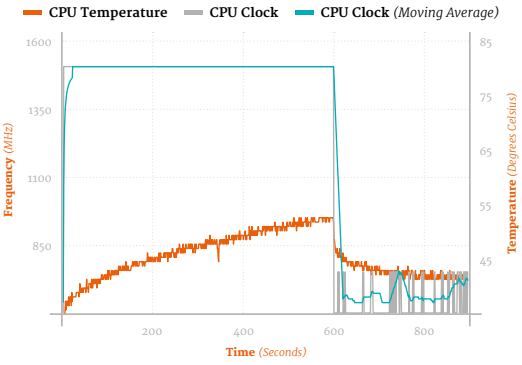
Thermal imaging

There's enough metal in the Argon One's aluminium upper shell to keep Raspberry Pi 4 cool even under sustained load.



Thermal load

After ten minutes of heavy load, the Argon One didn't even need to use its temperature-controlled fan once.



Verdict

There's little to fault with the Argon One's design. Cabling is tidied, the GPIO header made more readily accessible, and there's more than enough aluminium to keep Raspberry Pi 4 cool.

10/10

Sensoreq CooliPi Heatsink and Case

▶ Sensoreq ▶ **magpi.cc/coolipi** ▶ From £39 / $52

Featuring an open-source housing for a custom-milled heatsink, CooliPi is impressively extensible

SPECS

DIMENSIONS:
92.4×86×54.3 mm

MATERIAL:
Aluminium

WEIGHT (INC. ONE RASPBERRY PI 4):
320 g

NUMBER OF BOARDS SUPPORTED:
1

COOLING METHOD:
Passive heatsink (SoC, RAM, USB 3.0 controller), optional fan

EXTRAS:
Case 3D print files supplied

The CooliPi stands out from the competition not just owing to its size and weight – it's by far the heaviest case on test – but also by being at least partially open-source: while the custom-milled heatsink is available exclusively from Sensoreq, the plastic lower section can be printed on any 3D printer.

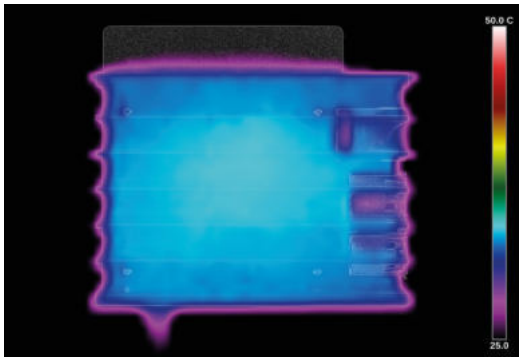
That's only part of the story. CooliPi is a family of products, not just a case, and optional extras – some of which are also 3D printable – include a 90-degree adapter for Raspberry Pi 4's GPIO header, a HAT mount, and even a housing for an optional 5 V fan.

> The CooliPi stands out from the competition by being at least partially open-source

The latter shouldn't be necessary outside the most extreme environments: in testing, the heavy heatsink of the CooliPi – which contacts the SoC, RAM, and USB 3.0 controller chips, with an optional copper shim available to cool the power management IC (PMIC) – was more than up to the job of cooling Raspberry Pi 4.

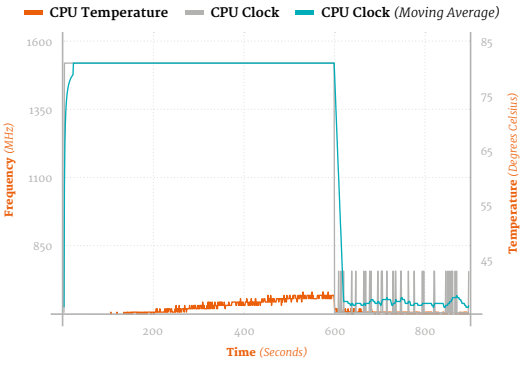
Thermal imaging

Having a very heavy aluminium heatsink lets the CooliPi absorb more heat than the competition.



Thermal load

The CooliPi's large heatsink made it by far the best-performing cooler in the group.



Verdict

The CooliPi can't be faulted on performance. Its price, however, is an issue: starting at £39/$52 for just the heatsink and case, it's the most expensive product on test.

8/10

Pimoroni Aluminium Heatsink Case for Raspberry Pi 4

▶ pimoroni.com ▶ **magpi.cc/heatsinkcase** ▶ £12 / $13.20

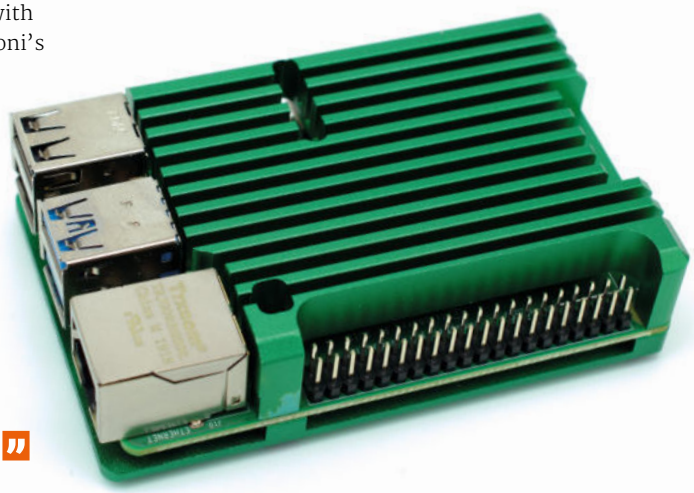
A compact two-part design, a few flaws don't stop this case performing well

A relatively straightforward two-part design, this all-aluminium affair aims to provide cooling and protection without taking up too much space – its overall footprint is only marginally larger than Raspberry Pi 4 on its own.

There are a few issues, though, starting with its design. Like all aluminium cases, Pimoroni's heatsink case includes pillars designed to contact hot-running chips and transfer the heat to the outside of the case. The installation instructions, however, tell you to only add a thermal transfer pad to the one in contact with the central SoC. It turns out that this is because the RAM pillar targets a chip which doesn't get hot, while the pillar for the USB controller is both too small and in the wrong place.

This, and a patchy anodised finish, aside, the case does as promised: it prevents Raspberry Pi 4 from throttling, and keeps all ports and headers – including GPIO, DSI, and CSI – readily accessible.

> "The pillar for the USB controller is too small and in the wrong place"



SPECS

DIMENSIONS:
88×56×22.4 mm

MATERIAL:
Aluminium

WEIGHT (INC. ONE RASPBERRY PI 4):
149 g

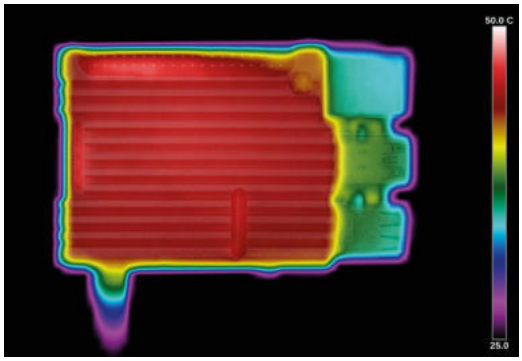
NUMBER OF BOARDS SUPPORTED:
1

COOLING METHOD:
Passive heatsink (SoC only, RAM and USB 3.0 controller optional)

EXTRAS:
Thermal transfer material pads, hex key

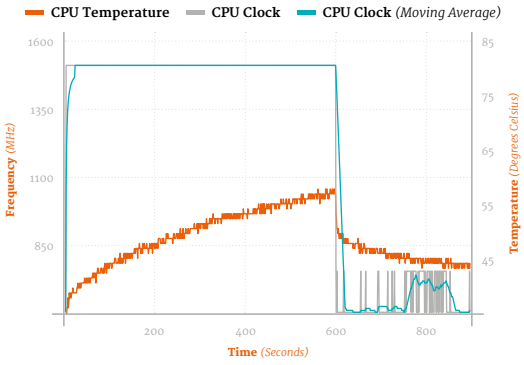
Thermal imaging

With so little metal to play with, the Pimoroni heatsink case gets noticeably warmer than the competition.



Thermal load

Even contacting only the SoC, the case keeps Raspberry Pi 4 well below its throttle point.



Verdict

The Pimoroni Heatsink Case does an acceptable job of cooling Raspberry Pi 4, but feels like a missed opportunity. Fixing the USB pillar and adding one for the PMIC would have been welcomed.

6/10

The Pi Hut
Raspberry Pi 4 Case 2.0

▶ The Pi Hut ▶ **magpi.cc/pihutpi4case** ▶ £10 / $11

A wholly acrylic creation, The Pi Hut's case relies on a small always-on fan to keep Raspberry Pi 4 cool

SPECS

DIMENSIONS:
97.7×69.7×36.3mm

MATERIAL:
Acrylic

WEIGHT (INC. ONE RASPBERRY PI 4):
125 g

NUMBER OF BOARDS SUPPORTED:
1

COOLING METHOD:
Fan

The Pi Hut's custom-designed Raspberry Pi 4 case comes in sheet form, laser-cut from a mixture of coloured and transparent acrylic. Assembly is relatively straightforward, though the plastic mounting pillars and screws provided can't withstand repeated assembly and disassembly, and there are no thermal interface pads required.

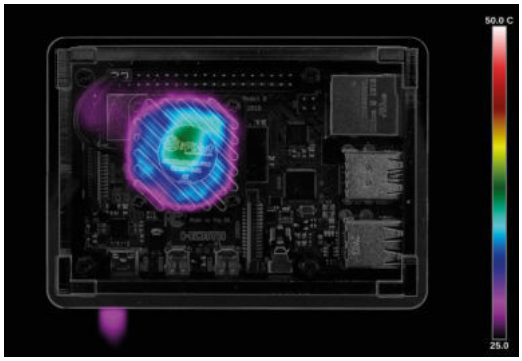
Instead, cooling is provided by a single 5V cooling fan installed beneath vents in the transparent lid. By default, this is set to suck air out of the case and away; flipping it around to blow offers a minor improvement in cooling performance at the cost of a dramatic increase in noise.

> There's no software or speed control for the fan, and it ties up the 5V and GND pins on the GPIO header

There's no software or speed control for the fan, and it ties up the 5V and GND pins on the GPIO header – which is inaccessible once assembled. The CSI and DSI headers are likewise locked away, though cables for these can at least be routed between the walls and the case lid.

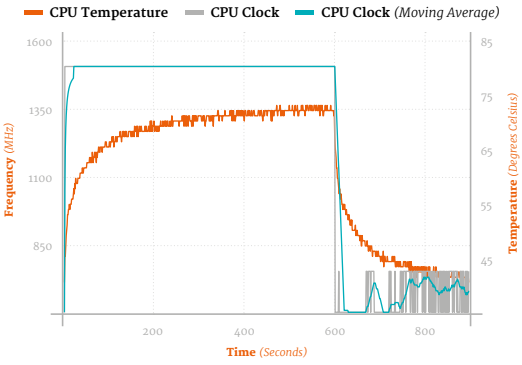
Thermal imaging

The acrylic lid effectively insulates Raspberry Pi 4, leaving the fan vent as the only place for heat to escape.



Thermal load

Despite its fan, The Pi Hut case's cooling performance is the second-worst on test – behind only the Raspberry Pi 4 Stand.



Verdict

The Pi Hut case is a cheap option. Despite including active cooling, it fails to outperform any of the passive options on test – bar only the in effect uncooled Raspberry Pi 4 Stand.

4/10

And The Winner Is...

Thermal performance isn't the be-all and end-all of choosing a case for Raspberry Pi 4 – in fact, as our testing in issue 88 proved, under most real-world workloads Raspberry Pi 4 is more than capable of handling itself.

It's little surprise, then, to find every case on test – except the Raspberry Pi 4 Stand – passed the demanding thermal throttle benchmark with flying colours.

> Any should prove more than adequate to prevent thermal throttling

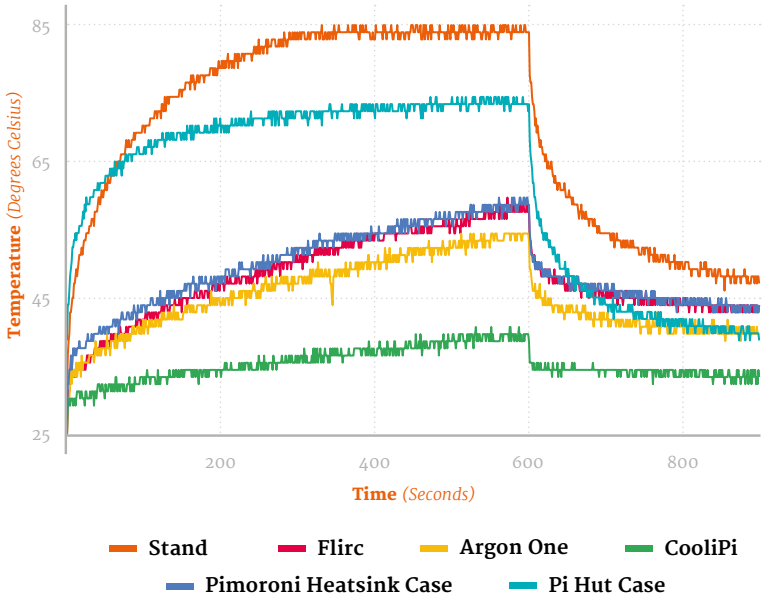
What is perhaps surprising is the variance within the tests. The Pi Hut case's fan isn't as effective as passive options like the Pimoroni Heatsink Case and the Flirc – and while the CooliPi is the best performer overall, its high price and bulk make for a difficult case to recommend for most use-cases.

Under real-world conditions, any of the cases – including the Raspberry Pi 4 Stand – should prove more than adequate to prevent thermal throttling. Only those operating Raspberry Pi 4 in relatively extreme environments need worry about cooling – and there's nothing wrong with picking your case based on features, accessibility, price, or aesthetics instead, opening up the whole group as potential winners depending on personal taste and budget.

Winner: Argon One

The Argon One ticks almost every box: it's attractive, includes a wealth of features, cools well, and won't break the bank.

Case cooling



Pirate Audio Headphone Amp

SPECS

AUDIO PROCESSING: PCM5100A DAC (24-bit / 192kHz), PAM8908 amplifier chip

DATA BUS: I²S for audio, SPI for LCD

DISPLAY: 240×240 IPS colour LCD

AUDIO OUT: 3.5mm stereo jack

CONTROLS: 4 × tactile buttons

▶ Pimoroni ▶ **magpi.cc/pirateaudio** ▶ £20 / $21

Create your own mini MP3 player with this DAC audio board, complete with an LCD to show track details. By **Phil King**

Raspberry Pi has its fair share of DAC audio boards offering high-quality sound output, but Pimoroni's new Pirate Audio range adds a mini LCD to show music track details and album art. In this review we're focusing on the Pirate Audio Headphone Amp, but we also tried out the other three models: Line-out, 3W Stereo Amp, and Speaker. We understand that another model is also set to be added to the range soon.

All the boards have the same slimline pHAT form factor that fits perfectly onto a Raspberry Pi Zero, although they'll work with any 40-pin model. The main difference between them is how the sound is processed and output. On the Headphone Amp, audio is amplified and then output via a 3.5mm jack – just plug in your wired headphones or earbuds. The positioning of the jack on the side of the board means you may need to take your Raspberry Pi out of its case, or raise it up using a booster header.

> It delivers excellent sound quality with a warm tone, plenty of fine detail, and sufficient bass for our ears

▶ Just plug in a pair of headphones or earbuds and enjoy your favourite music, from local storage or Spotify

Setting it up

Getting started with Pirate Audio wasn't quite as simple as we anticipated, although an online guide (**magpi.cc/pirateaudioguide**) has since appeared that should prove very helpful. Installing the default software itself is simple enough, by entering three commands in a Terminal window. This does everything needed to configure the DAC and enable SPI for the LCD.

Based around the Mopidy music server daemon (**mopidy.com**), the software enables you to play local music files or stream tracks from Spotify, although you'll need a premium account for that. The Spotify extension for Mopidy is installed automatically, along with one for the user-friendly Iris web interface.

The latter proves essential as you'll need to use it to actually start playing music on the board. Point a web browser to your Raspberry Pi's IP address appended with ':6680/iris' to access the web interface – you can do this from another computer or on the same Raspberry Pi if it's connected to a monitor.

The front of the board features four tiny tactile control buttons on either side of the colour LCD screen

Accessing local media files required a change to the Mopidy config file to reassign the local directory from the default to our Music folder, then running a local scan in Iris to find the files.

For Spotify streaming, you'll first need to authorise the device via the Mopidy website, alter the config file to enable Spotify and add your credentials, and then sign in again via Iris to start using Spotify from its interface. Tip: if you still get an error when trying to play files, try restarting the server from Iris's settings.

Music to our ears

The good news is that once you get everything set up, the Pirate Audio board's 24-bit, 192kHz DAC delivers excellent sound quality with a warm tone, plenty of fine detail, and sufficient bass for our ears. At first, we found it a bit too loud – until we flicked the switch on the rear of the board from high- to low-gain (recommended in most cases).

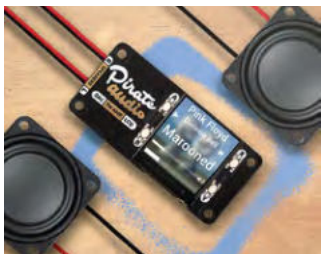
The volume level can be adjusted using two tiny control buttons on either side of the LCD. The other two buttons present are for play/pause and skip to next track in the queue, album, or playlist – there's no way of returning to previous songs unless you use the web interface.

We do love that built-in LCD, though, which shows you the track details on a background of the blurred album artwork, with a song progress bar at the bottom.

PIRATE AUDIO RANGE

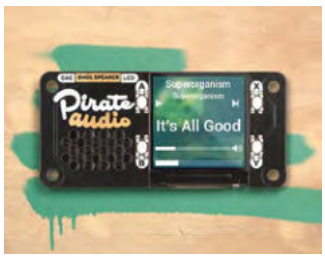
LINE-OUT

Aimed for use with powered speakers or by connecting to a hi-fi line input, it features line-level digital audio and a 3.5mm stereo jack.



3W STEREO AMP

This board features four tiny push-fit terminals on the rear to attach wires from passive speakers. There's also a switch for stereo and mixed-down mono modes.



SPEAKER

At only 1W, the small built-in speaker isn't very powerful and sounds rather tinny, but this board is ideal when you need integral audio for a portable project.

Verdict

A little tricky to get it all working, but once set up, the resulting audio is of excellent quality and the LCD is great for showing track details and artwork.

9/10

RetroFlag GPi Case

SPECS

DIMENSIONS:
138×81×32 mm

WEIGHT:
138 g

DISPLAY:
2.8-inch IPS

POWER:
3 × AA batteries or USB

COMPATIBILITY:
Raspberry Pi Zero or Zero W (not Zero WH)

▶ The Pi Hut ▶ **magpi.cc/gpicase** ▶ £60 / $79

PJ Evans takes retro gaming on the go with this eighties-handheld-inspired case with screen

If you're serious about retro gaming and would like to be able to play all the games in all the places, the RetroFlag GPi case might just be your new best friend. More than a simple enclosure for the Raspberry PI Zero, this rather familiar-looking unit is a full-blown battery-powered gaming console that (just about) fits in the palm of your hand.

Game on

The clever design of the GPi replicates that of eighties handheld consoles, including a cartridge-like unit that slides out of the rear and is quickly disassembled to reveal space for a Raspberry Pi Zero. A very neat 'pogo' mounting system means no header or soldering is required: a Raspberry Pi Zero just slots in and pressure maintains the contacts. The reassembled unit then slots into the main body and you're ready to go. You can even access the microSD card without removing the 'cartridge'. Power is supplied by three AA batteries or a supplied USB cable.

Full and clear instructions are provided to install support for the gamepad buttons and also the on/off switch that provides easy and safe shutdown. We found the unit easy to assemble and had RetroPie running in no time at all. We were particularly impressed by the screen, which uses IPS rather than TFT to give a razor-sharp display from any angle without any of the common side-lighting issues. A small audio speaker is built-in, with the option of headphones. The case itself is injection-moulded, solid, and beautifully made.

▼ Although bulky like classic 1980s handhelds, it feels comfortable in hand and the controls are well-placed

Verdict

This is one of the best gaming cases we've seen. A great design, easy assembly and, most of all, great fun to use. An essential purchase for any retro gamer.

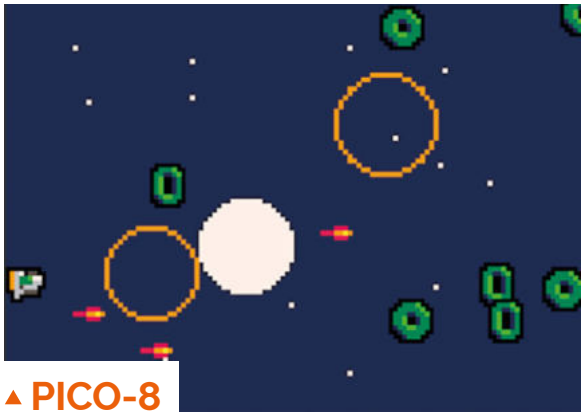
10/10

10 Best:

Game making projects

Learn how to make game software and hardware with these cool projects

When you have a bit of a coding hobby, you inevitably end up thinking about making your own game. Luckily, in this brand new decade, it could not be easier to find resources to help you get started. Here's how to begin making games with a Raspberry Pi.



▲ PICO-8

128 by 128 pixels of fun

Over several issues of *The MagPi*, Dan Lambton-Howard shows you how to use PICO-8 to create your own 8-bit wonder game made up of very limited pixels. Time to get creative.

magpi.cc/83

▲ **Interactive** tabletop RPG map

Roll for initiative

Upgrade your board games and tabletop games with this interactive map screen that can be used by a game master to gradually reveal dangerous dungeons or put up the next game board to play on.

magpi.cc/86

▼ **Build an** arcade cabinet

Learn how controllers work

What better way to learn how you can attach all the necessary peripherals – buttons, joysticks, speakers, lights, screens, etc. – to a Raspberry Pi for games than building a full arcade cabinet!

magpi.cc/63

▲ **Scratch Cat** goes skiing

Scratch-based gaming

Scratch is well suited to making games, with code scripts specifically tied to sprites/images. If you fancy wanting to learn the basics of coding while making a game in the process, check out this fun skiing game.

magpi.cc/catski

How to program games with the LÖVE gaming engine

The next step

LÖVE uses Lua, a popular language in game development, and this guide takes you through how it works in relation to Scratch – very useful for people just starting out with coding and game making.

magpi.cc/lovegame

3D robot Pac-Man

A reimagined classic

An oldie but a goodie, Pac-Man actually has AI for the individual ghosts, giving them unique movements/personalities. This version brings it to real life, adding an extra dimension in the process.

magpi.cc/66

▲ Flappy astronaut

Simple and educational

CCTV cameras with Raspberry Pi are not too difficult to put together. We like this project because it also uses computer vision and face detection to recognise known people.

magpi.cc/flappyastro

Make games with Python

Shoot-'em-up with Python

We put out an Essentials guide in 2015 which included a step-by-step guide to making a shoot-'em-up space game using Pygame, an amazing game programming library in Python.

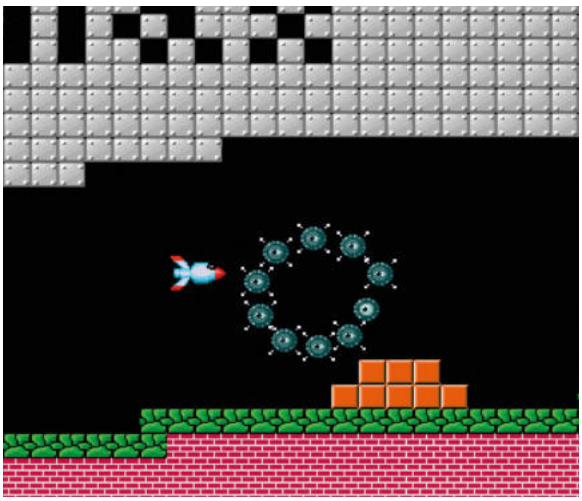
magpi.cc/gameswithpython

▲ Pymon

Simon says build this

This Simon clone device built by Les Pounder is a great example of a more physical game, and could even be used for the basis of a controller for a virtual game displayed on a screen.

magpi.cc/pymon



▲ Make games in C

Delve deeper into dev

A while ago we had a long-running series on coding games in C in *The MagPi*, starting in issue 65. Covering coding as well as basic game design theory, it was written by Brian Beuken, who teaches game programming at university.

magpi.cc/65

Learn Python with Raspberry Pi

Learning Python in 2020? **PJ Evans** catches up on the latest resources

EduBlocks

CREATOR

Joshua Lowe

Price: FREE

edublocks.org

As great a choice as Python is for your first adventure in coding, many start with Scratch (scratch.mit.edu), a fantastic free online coding environment that teaches the fundamentals of coding using visual 'blocks' to build up the set of instructions. Scratch is taught all over the world and is especially popular in schools. So what happens when it's time to transition to a more formal form of coding? Scratch can only take you so far, but the move to 'all text' Python can be daunting, especially to younger coders. Joshua Lowe agreed and (at the age of just twelve) created EduBlocks, which bridges the gap between visual block-based coding and Python. Simply create your Scratch-style program, click a single button, and pow, there's the equivalent Python code ready to cut and paste.

EduBlocks is online and completely free. If you're concerned about the next step in your coding adventure, this is the ideal place to play and experiment. You'll see that there isn't that much difference after all. The main site is supported by an extensive collection of tutorials and worksheets, many designed for Raspberry Pi.

Free Online Courses

Learning that won't hurt your wallet

YOUTUBE

Yes, it sounds obvious, but it's hard to overstate how many great resources are available on YouTube. The offerings from freeCodeCamp and Programming With Mosh are particularly recommended for Python beginners.

magpi.cc/youtubepython

DEEP LEARNING PREREQUISITES: THE NUMPY STACK IN PYTHON

Numpy (an extensive mathematical toolkit) was a game-changer, promoting Python to a major player in machine learning. This free course gives you a solid grounding in these advanced principles.

magpi.cc/udemynumpy

PROGRAMMING FOR EVERYBODY (GETTING STARTED WITH PYTHON)

How about learning about Python from the University of Michigan? This highly rated beginner's course features both tutorials and assignments, is available for free, and comes with online support from instructors.

coursera.org/learn/python

Learning IoT with Python and Raspberry Pi

CREATOR

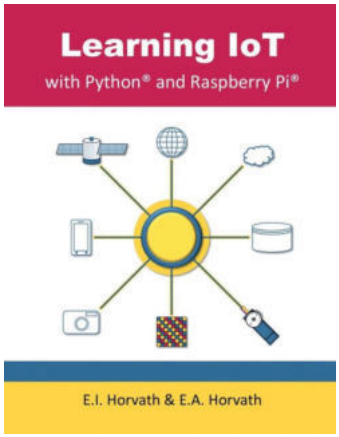
Dr EI Horvath & Dr EA Horvath

Price:
£95 / $125

learningiot.net

This impressive 760-page epic has been created by accomplished academics to provide a complete course that goes from your first line of code to creating real-life Internet of Things projects. Although it's intended to be picked up by academic institutions and implemented as a course, there's no reason not to self-teach if that's the style that works for you.

As well as a solid grounding in Python, from first principles to advanced techniques such as threading, its detailed projects include reading environmental data with Sense HAT, using GPS, and robot control. In addition to physical computing, there are many projects on interacting with the 'cloud', such as storing remote data and alerting.



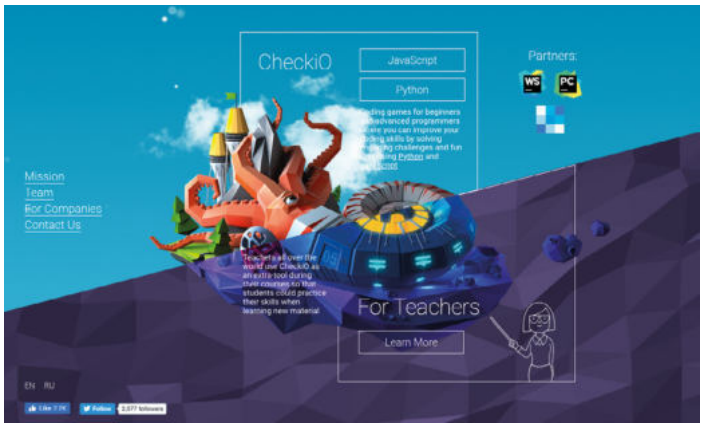
CheckiO

CREATOR

CheckiO

Price:
£10 / $10 per student (first class free)

checkio.org



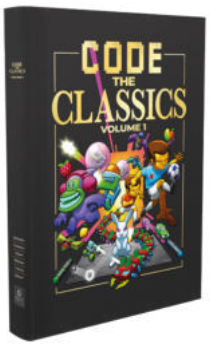
What's better than learning through gaming? CheckiO is a 100% online series of courses intended for the classroom, although individuals are also welcome. Through a series of over 200 tutorials, worksheets, and challenges, the participant starts with the fundamentals and writes more and more complex code as the principles of gaming are revealed. Throughout, there's an excellent support system, including the ability to discuss solutions with your peers. A scoring system and awards make this great fun for the classroom. If getting into the games industry appeals, or you just fancy creating some homebrew and don't know where to start, this might be the perfect choice.

Premium Python

Serious about Python knowledge? These are for you

PYTHON – FROM BEGINNER TO WINNER

This is a more structured course than others we've mentioned here. A total of three hours of video and 21 downloadable resources with coding exercises. Recommended for those wanting to learn in bite-size sections.
magpi.cc/udemypython



CODE THE CLASSICS

What could be better than learning through recreating the golden era of arcade gaming? This book (yes, from ourselves) teaches many fundamental principles of code while having some classic fun.
magpi.cc/ctc1

CODE COMBAT

Although aimed at the classroom, Code Combat is an epic collection of Python programming challenges wrapped in a sophisticated and fun user interface. You can play around for free.
codecombat.com

Pranjali Pathak

Bringing computer science to underprivileged children in India is a priority for this engineer turned educator

> Day job Program lead | > Community role Community support | > URL thepijam.org

Like a few educators in the Raspberry Pi and digital making community, Pranjali Pathak of the Pi Jam Foundation started out life as an engineer before becoming a teacher. She's been a fellow for Teach for India, where she taught students from grade 3, and brought that experience to the Pi Jam Foundation as a Program Lead, making sure educators are supported when teaching computing.

▼ Pranjali tries to make sure that girls get plenty of input to make up for the gender bias

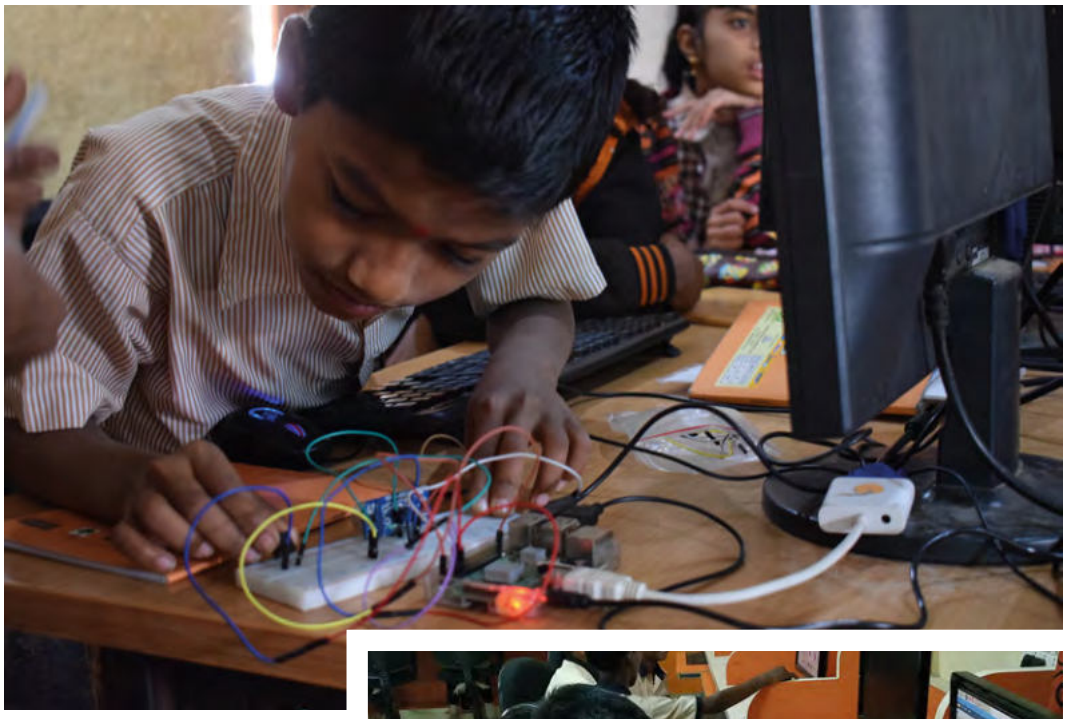
"Pi Jam Foundation is a Section 8 (not-for-profit) organisation," Pranjali tells us. "The organisation is entirely impact driven and aims to provide all students from under-resourced schools [with] computing and problem-solving skills, which are essential for them to succeed in the 21st century workplace. Using low-cost, open-source technology, Raspberry Pi and a contextual research-based curriculum coupled with innovative pedagogy, Pi Jam Foundation aims to provide quality computer education to over 100,000 students across India by 2022. We have successfully brought a globally relevant, yet contextual computer science education to 8000+ students across under-resourced schools across urban, suburban, and rural geographies of the country."

What are your links to Raspberry Pi?

We work extensively on Raspberry Pi [computers] across all our Pi Lab programs and feel the Code Clubs would enable us to enhance our curriculum and provide students from different grades and diverse economic backgrounds access to knowledge at par with global standards. Also, being [more] affordable than existing PCs in the market lends itself to use in the context we operate in, i.e. under-resourced public schools.

What kind of events have you put on/supported?

Pi Lab is an award-winning flagship programme that enables a complete computer science ecosystem (Raspberry Pi, best of open-source tools, and highly contextualised curriculum and teacher training



People from all backgrounds attend the events

programmes) that ensures year-long learning and allows kids to explore, tinker, and create. Currently, Pi Jam is impacting 8000+ number of students through 40 Pi Labs across [...] the country.

The Pi Labs, especially the ones in under-resourced schools, have been places of immense transformation of thought and possibility for our students. They have seen technology and computer science as something the more privileged classes had access to, and using Raspberry Pi [computers] has helped us give them the same (if not a more focused and holistic) access.

We see career possibilities change as they experience the true creativity behind technology, and have even encountered them sharing with us that they always assumed technology to be boring, but the Code Clubs were very exciting for them to be a part of.

Annual Showcase 'Makers Factory': Pi Jam's Annual Showcase takes place every February, where our students are provided with a platform to present their solutions in the form of technology prototypes to problems identified by them.

> Pi Lab is an award-winning flagship programme

Hackathons: for students, members from the community, mothers of the Pi Lab students, [and] individuals from corporates.

What kind of people attend these events?

The students hail from communities that have a rich mix of people from different cultures, beliefs, and languages. The lack of resources and rampant poverty unite residents of this locality. Most of the communities are home to tens of thousands of children, many of whom lack access to a good-quality education.

All the students come from low-income households, and a majority of their parents are first-generation migrants from other states. Additionally, most of the students do not have any formal knowledge or prior experience to computers across these schools.

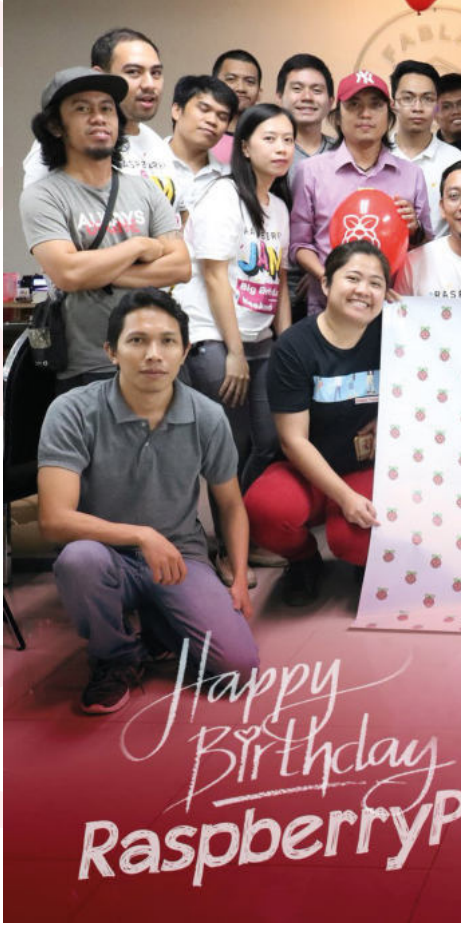
Pi Jam projects

Rotten fruit detector

Rukaiya's parents were fruit and veg sellers; however, a problem with the vendor meant some fruit was rotting and affecting the rest of the supply. Unable to inspect the boxes by eye, Rukaiya created a rotten fruit detector, which used a stick with an MQ3 sensor on the end which would smell out rotting fruit in the box.

Open manhole alert

During the monsoon season, it's not uncommon for autos (motorised rickshaws) to get stuck in manholes. Several students had to abandon a taxi auto at the urge of their driver and felt bad for him, so decided to create a system that warned the driver in the future of open manholes in the road ahead. It's incredibly accurate as well, even being able to detect if a manhole is slightly tilted open.



This Month in Raspberry Pi

Big birthday list!

Here are just some of the Raspberry Pi events celebrating its eighth birthday

As we mentioned last issue, Raspberry Pi is turning eight this year! Or two, if you only count leap years. Either way, in honour of this birthday, any Raspberry Jams taking place between Saturday 15 February and Sunday 15 March can be used to celebrate Raspberry Pi!

Here are the events that have been planned by the time we go to print. For a more up-to-date list, head to **rpf.io/jam**.

UK

BLC Raspberry Jam
Buxton

Beeston Raspberry Jam
Nottingham

CamJam
Cambridge

South Devon Tech Jam
Paignton

Chelmsford Raspberry Jam
Chelmsford

Cornwall Tech Jam
Redruth

Exeter Raspberry Jam
Exeter

Sheffield Raspberry Jam
Sheffield

Gateshead Raspberry Jam
Gateshead

Southend Raspberry Jam
Southend-on-Sea

Leeds Raspberry Jam
Leeds

York Big Birthday Pi Jam
York

London Raspberry Pint
London

USA

Raspberry Pi Birthday Jam @ California High School
California

Fresno Ideaworks Pi Day Party
California

Raspberry Jam Catoosa
Georgia

Jelly's Raspberry Jam Party
Maryland

Raspberry Pi Big Birthday Jam Ann Arbor
Michigan

Los Alamos Raspberry Jam
New Mexico

TEC Raspberry Jam
Tennessee

Raspberry Pi 8th Anniversary Celebration!
Utah

Roanoke Raspberry Jam
Virginia

Columbia Basin Jam
Washington

Raspberry Jam WV
West Virginia

Europe

Raspberry Jam Zelzate
Belgium

Varna Raspberry Jam
Bulgaria

Raspberry Jam des inspirés 2020
France

Votanikos Jam
Greece

AmsterJam
Netherlands

CoderDojo Baarle Raspberry Birthday party
Netherlands

PiWars
Turkey

Rest of the World

Litoral Raspberry Jam
Argentina

Raspberry Pi JAM Buenos Aires
Argentina

Santiago del Estero, Argentina Raspberry Jam
Argentina

Inverell Raspberry Jam
Australia

Calango Raspberry Jam
Brazil

Raspberry Jam São Paulo
Brazil

Pi Jam @ PiLabs
India

Raspberry Pi Jam Kochi
India

Abhinav Raspberry Jam
India

Atria Labs Raspberry Jam
India

Bhopal Raspberry Jam
India

PiWorld Raspberry Pi Jam
India

ResPro Labs Raspberry Pi Jam
India

Pi Jam 2020 @ SRiX Warangal
India

Raspberry Jam NextTech
India

Itierio Boys' High School Raspberry Jam
Kenya

Perak Technology Academy: Raspberry Pi Jam
Malaysia

Jam Kathmandu
Nepal

Ikeja Tech Mates Raspberry Jam
Nigeria

Pi Day with Free City Hacks
Nigeria

Davao Raspberry Birthday Jam
Philippines

Suñu Raspberry Jam
Senegal

Bangkok Raspberry Jam
Thailand

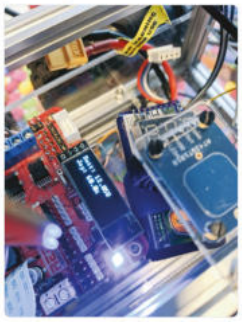
Dubai Raspberry Birthday Jam
United Arab Emirates

MagPi Monday

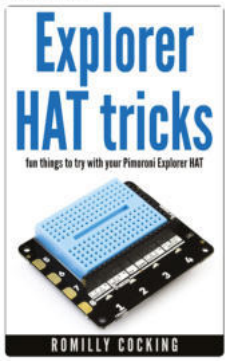
Amazing projects direct from our Twitter!

Every Monday we ask the question: have you made something with a Raspberry Pi over the weekend? Every Monday, our followers send us amazing photos and videos of the things they've made. Here is a small fraction of them. Follow along at the hashtag #MagPiMonday.

01



02



03

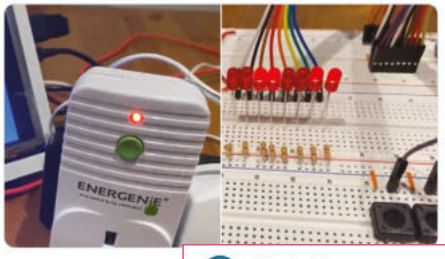
orionrobots
@orionrobots
Replying to @TheMagPi
Hello, Ive been 3d printing a mount to link the Raspberry Pi camera with Lego technic (for a robot of course), and mentoring Raspberry Pi makecode Arcade at @CoderDojoHam with @rdhayler .

04

Michael (Mike) Horne
@recantha
Replying to @TheMagPi
I've been testing out the AmbiMate MS4 sensor package from @TEConnectivity te.com/usa-en/product ... and finding out about the perils of I2C clock stretching recantha.co.uk/blog/?p=19880

05

Mark Bush
@BikrSpeedTriple
Replying to @TheMagPi
Implementing the C64 user port to control electronics in BASIC in the amazing #BMC64 app. (Also for C128, VIC20, and PET!)



06

8 Bits and a Byte
@8BitsandaByte
Replying to @TheMagPi
An internet controlled monster :)
Translate Tweet

07

Dr Footleg - Roboteer
@drfootleg
Replying to @TheMagPi
I made my fastest Raspberry Pi robot yet:

Dr Footleg - Roboteer @drfootleg · Jan 1
First robot build of 2020. I present Turbo Tiny. A test bed chassis for my robot HAT.

01. This looks like a scene from *Short Circuit*

02. Good luck with the book!

03. 3D printers make custom robot work so very easy

04. Clock stretching is a tricky bit of code magic

05. Good use of an Energenie

06. This monster looks familiar. Crumbs, we can't place it…

07. The image doesn't do this justice: it is extremely fast

Best of the rest!

Here are some other great things we saw this month

SANIA BOX

This is a Kickstarter for a special add-on board for Raspberry Pi that is aimed at teaching code to eight-year-olds, with an initial idea coming from a 13-year-old. The add-on board has LEDs, a push-button, thermal sensor, touch sensor, light sensor, and an environmental sensor for gases.

▶ **kck.st/38fj9pd**

NOVELTY RADIO PC

We think this is very cute – Reddit user carecoin took a tiny novelty radio and put a Raspberry Pi Zero W and screen into it. There's no banana for scale, but it's very small.

▶ **magpi.cc/noveltypc**

CROWDFUNDING A PROJECT?

If you've launched a Raspberry Pi-related project, let us know!

magpi@raspberrypi.org

Raspberry Jam Event Calendar

Find out what community-organised Raspberry Pi-themed events are happening near you...

01. Exeter Raspberry Jam

- Saturday 1 February
- **Exeter Library, Exeter, UK**
- **magpi.cc/jM5Qp7**

A meeting for everyone interested in all things microcomputers, microcontrollers, robotics, and making.

02. St Louis Raspberry Pi Jam

- Saturday 1 February
- **Danforth Center, St Louis, MO, USA**
- **magpi.cc/HunpKi**

An event where makers can work in a collaborative group on a project, or just learn about makers and making.

03. Melbourne Raspberry Pi Hackers

- Sunday 2 February
- **Library at The Dock, Melbourne, VIC, Australia**
- **magpi.cc/cHa3y8**

A meet-up open to everyone with an interest in electronics, robotics, making, 3D printing, laser cutting, etc.

04. Barnstaple Raspberry Jam

- Saturday 8 February
- **Barnstaple Library, Barnstaple, UK**
- **magpi.cc/sVtkkf**

A friendly bunch of Raspberry Pi enthusiasts who meet once a month to share ideas. All ages and abilities welcome.

05. Cornwall Tech Jam

- Saturday 8 February
- **Bodmin Library, Bodmin, UK**
- **cornwalltechjam.uk**

For anyone interested in technology, of all ages and abilities. Ask questions and learn about programming.

06. Manchester Raspberry Jam

- Saturday 8 February
- **The Shed, Manchester, UK**
- **magpi.cc/hyNWr8**

A monthly user group where attendees can take part in taught workshops or get help working on their own projects.

07. South Devon Tech Jam

- Saturday 8 February
- **Paignton Library and Information Centre, Paignton, UK**
- **magpi.cc/9vhGQ5**

A monthly informal and friendly session for anyone interested in technology, regardless of age or ability.

08. Birthday Jams!

- Saturday 29 February
- **Earth, Sol System, Milky Way**
- **rpf.io/jam**

Raspberry Pi turns eight this year! To celebrate, there will be Jams around the world for a month! See page 84 for more

FULL CALENDAR

Get a full list of upcoming events for February and beyond here:

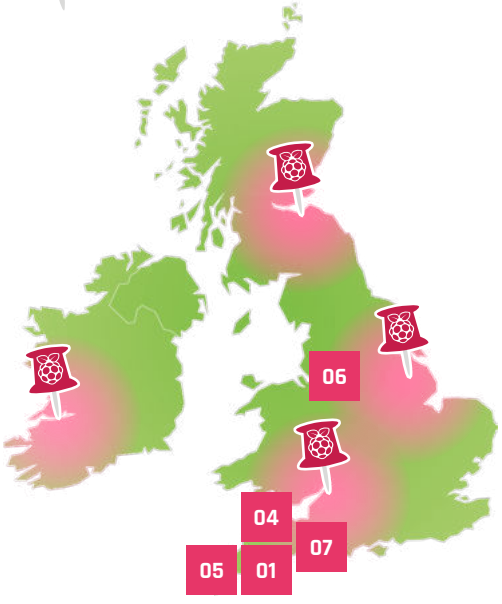
rpf.io/jam

FIND OUT ABOUT JAMS

Want a Raspberry Jam in your area? Want to start one?

jam@raspberrypi.org

We've highlighted some of the areas in need of a Jam! Can you help out?



03

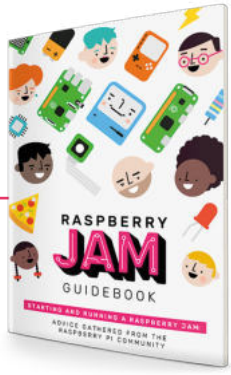
Raspberry Jam advice:

Kit

"When I ran my first Jam in 2012, you could only buy one Raspberry Pi at a time, and you had to wait about three months. The Jam was essentially the 30 people in Manchester who were lucky enough to buy one on day one! Following that, more people managed to get hold of a Raspberry Pi. It wasn't a problem that the Jam had no kit of its own: people were expected to bring their own. They were happy working on their own personal projects, and getting help from each other."

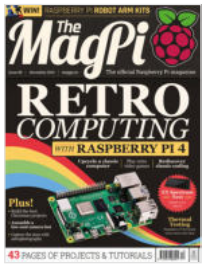
Ben Nuttall – Manchester Raspberry Jam

Every Raspberry Jam is entitled to apply for a Jam starter kit, which includes magazine issues, printed worksheets, stickers, flyers, and more. Get the book here: **rpf.io/guidebook**




FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE
RETRO GAMING WITH RASPBERRY PI
RETRO GAMING
WITH
RASPBERRY PI
164 PAGES OF
VIDEO GAME PROJECTS
+ BUILD AN ARCADE MACHINE
PLAY & CODE GAMES!
AMAZE BALLS!

Your Letters



▲ There are many places that would take copies of *The MagPi* and HackSpace magazine

Donating issues

Anyone want a load of *The MagPi* and HackSpace magazine issues? I need the room and I really don't want to just bin them. We don't have room for them at school either.

Darren via Twitter

Unfortunately, this is not an easy one to solve! We've retweeted your tweet, but that's about as much as we can do – we do suggest that you look for any local Raspberry Jams, Code Clubs, CoderDojos, or makerspaces and get in contact with the organisers to see if they would like them. Otherwise, please recycle them responsibly!

51st tip

Great issue [Ed – in reference to issue 89, magpi.cc/89]: I particularly like Dale's Scratch tutorial on p44. Referring to your tips feature on p26 – one thing I find really handy is something that insulates. I was told to use clear nail varnish. Is this the best form of insulator or can you recommend something else?

Anne via email

▲ Do you have any extra tips you'd like to share? Email us!

Using clear nail varnish is not something we've heard before, but apparently it is completely real as a valid non-conductive insulator. It seems a few folks like to use flat colours like red and black for colour-coding if they need it as well. We reckon if you want to really do something fancy, check out UV varnish.

As for other insulators, we like to heat up a glue gun and use the low setting to cover anything that might short. It's especially effective for costume electronics, as it helps to keep wires attached if they're moving more than they should.

Contact us!

- Twitter **@TheMagPi**
- Facebook **magpi.cc/facebook**
- Email **magpi@raspberrypi.org**
- Online **raspberrypi.org/forums**

Coolest Projects registration

I read a couple issues ago that registration for the American and UK versions of Coolest Projects was open, but registration for the International one wasn't yet. I assume it's because it's a bit later in the year, but do you have any idea when we might learn news of its opening?

Serg via Facebook

You're in luck – Coolest Projects International registration opened just before we went to print! It's being held in Dublin on 6 June, and it's available for everyone around the world. If you want to check out more about Coolest Projects, and register for the International event, head here: **coolestprojects.org/international.**

▼ We love to see some of the amazing projects that come out of Coolest Projects events!

◀ The official Raspberry Pi 4 power supply is a great charger for USB-C devices

Power supply upgrade

I'm thinking of upgrading to a Raspberry Pi 4 this year and was wondering what accessories I'd also need to upgrade to use it properly. I'm currently using a Raspberry Pi 3B, and have all the standard kit for that. I know I'll need a new case and a micro HDMI cable, but otherwise am I good to go? I know that sometimes a power supply upgrade is needed, will I need a new PSU?

Jon via email

The power port of Raspberry Pi 4 is USB-C, while older Raspberry Pi models use micro USB 2.0, so you will need some kind of upgrade to your power solution.

Now, you don't necessarily need to get a USB-C charger – you can get a micro USB to USB-C adapter if your current power supply is powerful enough. If it's about 3A or higher, you should be OK.

Otherwise, if you've got a new Android phone over the last year or so, you may already have a USB-C PSU. These are generally powerful enough to run Raspberry Pi 4 with few issues. Of course, if you don't have a charger, you can grab the official power supply.

CUSTOM PC

THE BEST-SELLING MAG FOR PC HARDWARE, OVERCLOCKING, GAMING & MODDING

THE MAGAZINE FOR

PC HARDWARE ENTHUSIASTS

ISSUE 198 OUT NOW

VISIT **CUSTOMPC.CO.UK** TO LEARN MORE

WIN ONE OF FIVE PROTODOCK ADD-ONS!

Sometimes you have just a few too many components, microSD cards, and other gizmos for your Raspberry Pi projects. The ProtoDock helps you organise them.

> The ProtoDock keeps your desk tidy and all of your vital Raspberry Pi accessories in one place

Note: Raspberry Pi and accessories, other than breadboard, not included

Head here to enter: magpi.cc/win | **Learn more: magpi.cc/protodock**

Terms & Conditions

Competition opens on **29 January 2020** and closes on **27 February 2020**. Prize is offered to participants worldwide aged 13 or over, except employees of the Raspberry Pi Foundation, the prize supplier, their families, or friends. Winners will be notified by email no more than 30 days after the competition closes. By entering the competition, the winner consents to any publicity generated from the competition, in print and online. Participants agree to receive occasional newsletters from The MagPi magazine. We don't like spam: participants' details will remain strictly confidential and won't be shared with third parties. Prizes are non-negotiable and no cash alternative will be offered. Winners will be contacted by email to arrange delivery. Any winners who have not responded 60 days after the initial email is sent will have their prize revoked. This promotion is in no way sponsored, endorsed or administered by, or associated with, Instagram or Facebook.

HackSpace

TECHNOLOGY IN YOUR HANDS

THE NEW MAGAZINE FOR THE MODERN MAKER

SUBSCRIBE AND SAVE UP TO 35% on the cover price

ISSUE #27

OUT NOW

hsmag.cc

Available on the App Store

GET IT ON Google Play

THE MAGPI #91
ON SALE 27 FEBRUARY

Plus!

Raspberry Pi reaching robot

Build a portable arcade machine

Detect motion with Raspberry Shake

DON'T MISS OUT!

magpi.cc/subscribe

TWITTER	@TheMagPi
FACEBOOK	fb.com/MagPiMagazine
EMAIL	magpi@raspberrypi.org

EDITORIAL

Editor
Lucy Hattersley
lucy@raspberrypi.org

Features Editor
Rob Zwetsloot
rob.zwetsloot@raspberrypi.org

Sub Editors
Phil King and Nicola King

ADVERTISING

Charlotte Milligan
charlotte.milligan@raspberrypi.org
+44 (0)7725 368887

DESIGN

criticalmedia.co.uk

Head of Design
Lee Allen

Designer
Sam Ribbits

Illustrator
Sam Alder

CONTRIBUTORS

Alex Bate, Mike Cook, David Crookes, PJ Evans, Tracy Gardner, Gareth Halfacree, Rosie Hattersley, Nicola King, Simon Long

PUBLISHING

Publishing Director
Russell Barnes
russell@raspberrypi.org

Director of Communications
Liz Upton

CEO
Eben Upton

DISTRIBUTION

Seymour Distribution Ltd
2 East Poultry Ave,
London EC1A 9PT
+44 (0)207 429 4000

SUBSCRIPTIONS

Unit 6 The Enterprise Centre
Kelvin Lane, Manor Royal,
Crawley, West Sussex, RH10 9PE
+44 (0)1293 312193
magpi.cc/subscribe
magpi@subscriptionhelpline.co.uk

This magazine is printed on paper sourced from sustainable forests and the printer operates an environmental management system which has been assessed as conforming to ISO 14001.

ISSN: 2051-9982.

Do-it-yourself computing

Why DIY is about more than saving a few pennies. By **Lucy Hattersley**

As all my friends can testify: I love DIY. Give me a drill and a wall to damage and I'm a happy gal.

And my DIY cup runneth over, since I recently moved into a new house that our local plumber Naim cheerfully referred to as: "Oh my God Luce this place is a wreck, are you sure?"

The estate agents didn't even say it had 'character'. The brochure said, with a hopeful look, that our new home was "in need of cosmetic decoration" and presented a "fantastic opportunity to add value to a property".

And add value I have! Day and night with drill, and hammer, and screwdriver. The cat hates me; carpets have come up, stairs have been pulled down, and with more than a little help from Naim, a new kitchen has magically appeared.

And all this DIY has got me thinking about Raspberry Pi.

How does it work?

Raspberry Pi is – above all else – the do-it-yourself of computing platforms. Not for us the glue and glass of the iPhone and the security screws of the Microsoft Surface. Oh no, we hold our computer in our hand and say out loud, "How do you work, then, little one?"

Sometimes we also say, "Why aren't you working, you absolute rotter." [We don't swear in *The MagPi*, house rule – Ed.]

Doing-it-myself, or rather undoing the work of others to redo it, has given me an appreciation of good work. Good work can be taken apart, examined, understood, and reassembled. Raspberry Pi is good work.

> "I want to live in a world that I understand. If something goes wrong, I know how to fix it"

And I think doing-it-yourself is important. Because the alternative is somebody-else-does-it-for-you. And that comes at a heavy cost (and I don't mean money). Somebody-else-does-it-for-you means they get to understand the world you live in. They get all the joy of a job well done; if something goes wrong, you need to call them and ask for help.

I want to live in a world that I understand. If something goes wrong, I know how to fix it, how to stop it getting worse, and at least where to look to solve the problem.

Do-it-yourself computing is important. In a world of locked-down devices and black-box artificial intelligence, do-it-yourself computing is a shield against superficiality. It helps you to take the modern world apart and get a grip on how it ticks.

It's also good for the environment. Learn to make something and you can fix and repair it, improve it, and gain an appreciation for the immediate world around you.

Sustainable devices

Repair is the new green, as iFixit says (**magpi.cc/ifixitrepair**): "Sustainable devices are devices that are designed to have a long life – and that means prioritising repair and upgradability. Devices that are easier to disassemble are also easier to recycle. Reusing, repairing, and recycling are always greener than manufacturing something anew."

DIY – both in your computing and in your home – is good for the soul and good for the world. Long may it continue.

AUTHOR

Lucy Hattersley

Lucy is the editor of *The MagPi* and is doing it all just to get more space for a dog. A dog with a lot to live up to.

magpi.cc

Universal & Unique

UniPiCase

for Raspberry Pi 4

Perfect option for your DIY or high-volume commercial projects

Fits all kinds of HATs
Ideal PoE HAT case

Use with HAT or Pi alone
Simple and professional

Rapid, tool-free assembly
GPIO cable pass-through

Two wall mount options
Well ventilated

www.UniPiCase.com

Designed and manufactured in North America

PIPER

INSPIRE FUTURE GENERATIONS
WITH PIPER COMPUTER KIT 2

the award-winning computer kit for ages 8+

Find out more about this and many other products at

www.okdo.com